UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

THÈSE

PRESENTÉE À
UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

EN VUE DE L'OBTENTION DU DIPLÔME DE
DOCTORAT EN ADMINISTRATION – GESTION DE PROJET

IN PARTIAL FULFILLEMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTORATE IN ADMINISTRATION – PROJECT MANAGEMENT

JOSEPH POTVIN

DATA WITH DIRECTION:

DESIGN RESEARCH LEADING TO A SYSTEM SPECIFICATION FOR ‘AN INTERNET OF RULES’

GATINEAU, QC — 2023-01-01

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Hamed Motaghi, Jury Chair
Département des sciences administratives
Université du Québec en Outaouais

Nhamoinesu Mtetwa, External Evaluator
Data Science Engineering Manager, Barclays
Formerly Lecturer and Articulation Lead,
School of Computing, Engineering and Built Environment
Glasgow Caledonian University (2014-2020)

Raphaël Khoury, Internal Evaluator
Département d'informatique et d'ingénierie
Université du Québec en Outaouais

Stéphane Gagnon, Thesis Supervisor
Département des sciences administratives
Université du Québec en Outaouais

THIS THESIS WAS PRESENTED AND DEFENDED ONLINE

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC ON

12 January 2023

HOSTED BY UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

Abstract

(Un résumé et une synthèse de la thèse en langue française suivent ci-dessous.)

This research fills a gap in project management theory and practice, which concerns how any stakeholder discovers and obtains factual knowledge of the significant rules that are ‘in effect’ for dates/times and prerogatives relating to identities and jurisdictions of a given context; that are ‘applicable’ to the class of endeavour and task being undertaken; and that are ‘invoked’ by a particular circumstance of the moment.

Something has to supply the directionality to projects. Rules are directional relations between what ‘is’ and what ‘ought’ to be, established among two or more individuals or entities. In practical, logical, ethical and aesthetic matters, rules express obligation, permission or encouragement through the commonly capitalized terms MUST, MAY and SHOULD, or their various negatives and synonyms.

This dissertation excavates the conceptual foundations of rules and rule systems, and describes the rationale, design, feasibility, generalizability and utility of a networked computational method for anyone to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate normative data which relates what ‘is’ with what ‘ought’ to be, with deference to prerogatives, agreements and preferences. We propose the Data With Direction Specification (DWDS) as a specification for a class of data-processing pipeline summarized with the relation 'IS + RULE ⟾ OUGHT'.

Applied research has been pursued concurrently for proof-of-concept validation through the development of working reference implementation software under free-libre-open source licensing and methods. This has included active peer review of the iterative design with professionals in a variety of potential implementation communities. The reference implementations are minimum working models of the specifications, including a RuleMaker application, a RuleTaker component, and a RuleReserve network service. Running these together enables an emergent ‘Internet of Rules’. Sample use cases other than those which illustrate various functional details are beyond the scope of the present design research which is to enable a general purpose system.

This dissertation begins with a problem statement, a multi-faceted ‘available methods review’ (which is akin to a literature review, but is focused on functional design), and a comprehensive design research’ methodology. The core of the dissertation is a precise technical rationale and design description of a decentralized distributed network service for anyone to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate rules that are ‘in effect’, ‘applicable’ and ‘invoked’ by a circumstance, across any informatics network, with precision, simplicity, scale, speed and resilience, along with deference to prerogatives, agreements and preferences.

Résumé

(The dissertation in English begins on page 39.)

Cette recherche comble une lacune dans la théorie et la pratique de la gestion de projet, qui concerne la façon dont toute partie prenante du projet découvre et obtient une connaissance factuelle des règles significatives qui sont « en vigueur » pour les dates/heures et les prérogatives relatives aux identités et aux juridictions d'un contexte donné ; qui sont « applicables » à la catégorie d'effort et de tâche entreprise ; et qui sont « invoquées » par une circonstance particulière du moment.

Le concept original le plus essentiel apporté dans cet ouvrage est exprimé dans le formalisme :

$$\text{SOIT} + \text{REGLE} \Longrightarrow \text{DOIVE}$$

Cette relation est, bien sûr, intrinsèque à la littérature philosophique sur les règles, mais elle n'a pas été exprimée auparavant de manière aussi succincte.

Il faut que quelque chose fournisse la directionnalité aux projets. Les règles sont des relations directionnelles entre ce qui « est » et ce qui « devrait » être, établies entre deux ou plusieurs individus ou entités. Dans les domaines pratique, logique, éthique et esthétique, les règles expriment l'obligation, la permission ou l'encouragement par le biais des termes couramment utilisés en majuscules DEVOIR, PEUT et SOUHAITER, ou de leurs divers négatifs et synonymes.

Cette thèse explore les fondements conceptuels des règles et des systèmes de règles, et décrit le raisonnement, la conception, la faisabilité, la généralisation et l'utilité d'une méthode informatique en réseau permettant à quiconque d'écrire, de rédiger, publier, découvrir, obtenir, scruter, prioriser et, avec l'accord des parties prenantes directes, automatiser des données normatives qui relient ce qui « est « à ce qui « devrait « être, en respectant les prérogatives, les accords et les préférences. Nous proposons le Data With Direction Specification (DWDS) comme une spécification pour une classe de pipeline de processus de données résumée par la relation SOIT + REGLE ⟹ DOIVE.

La recherche appliquée a été poursuivie en parallèle pour valider le concept par le développement d'un logiciel de mise en œuvre de référence fonctionnel sous licence et méthodes libres. Cela a inclus une révision active par les pairs de la conception itérative avec des professionnels de diverses communautés de mise en œuvre potentielles. Les mises en œuvres de référence sont des modèles de travail minimaux des spécifications, y compris une application RuleMaker, un composant RuleTaker et un service réseau RuleReserve. L'exécution de ces éléments ensemble permet l'émergence d'un « Internet des règles ». Les exemples de cas d'utilisation qui ne sont pas ceux qui illustrent divers détails fonctionnels dépassent le cadre de la présente recherche de conception, qui vise à mettre en place un système à usage général.

Cette thèse commence par un résumé du problème, une « revue de l'art antérieur » à plusieurs facettes (qui est comme une revue de la littérature, mais qui se concentre sur la conception fonctionnelle), et une méthodologie complète de « recherche de conception ». Le cœur de la thèse est un raisonnement technique précis et une description de la conception d'un service de réseau distribué décentralisé permettant à quiconque d'écrire, de rédiger, publier, découvrir, obtenir, scruter, prioriser et, avec l'accord des parties prenantes directes, automatiser des règles qui sont « en vigueur », « applicables » et « invoquées » par une circonstance, à travers n'importe quel réseau numérique, avec précision, simplicité, échelle, vitesse et résilience, tout en respectant les prérogatives, les accords et les préférences.

Synthèse de la thèse en français

(The dissertation in English begins on page 39, following this French language summary.)

Chapitre 1 : Introduction

1.1 L'objectif

Cette recherche remplit une lacune dans la théorie et la pratique de la gestion de projet, qui concerne la manière dont une partie prenante du projet est supposée de découvrir et d'obtenir une connaissance factuelle des règles significatives qui sont « en vigueur » pour les dates/heures et les prérogatives relatives aux identités et aux juridictions d'un contexte donné ; qui sont « applicables » à la catégorie d'effort et de tâche entreprise ; et qui sont « invoquées » par une circonstance particulière du moment. Les personnes pratiques, en tant qu'individus et au nom d'entités, doivent avoir une connaissance opportune et compréhensible des règles pertinentes afin d'exercer leur jugement pour s'aligner activement sur elles ou décider de ne pas le faire.

Une « règle » est un canon ou un précepte par lequel un comportement répété est guidé par l'autorité, l'accord ou la préférence. Les règles sont exprimées sous forme d'assertions avec les verbes auxiliaires modaux shall (doit), should (devrait) ou may (peut). Tout agent humain qui entreprend un projet est à la fois un créateur de règles, et toute entreprise humaine est façonnée par des règles explicites. Mais il n'existe aucun moyen commun, systématique et efficace de communiquer l'obligation, la permission ou l'encouragement de ceux qui fixent les règles à ceux qui les appliquent. Ce moyen est nécessaire pour assurer la cohérence, la fiabilité et l'efficacité de la prise de décision à base de règles tout au long des chaînes de valeur des projets.

Les données normatives exprimant DOIT, PEUT et DEVRAIT, ainsi que leurs divers négatifs et synonymes, sont structurées dans ce système comme une classe distincte de données avec une directionnalité intrinsèque qui peut être instantanément découverte et transmise sur Internet, sous une forme directement utilisable par des humains et des machines non spécialisés, pour n'importe quel objectif, dans n'importe quelle langue. Il s'agit d'une solution universelle au problème de la communication de la direction à prendre pour orienter les décisions impliquant de nombreuses personnes, dans les micro-décisions en cours pour la gestion quotidienne des projets, des programmes, des portefeuilles ou des plates-formes, et dans les initiatives, les mécanismes et les structures de conception des systèmes au niveau macro.

Nous nous intéressons ici à la conception d'une méthode fournissant à toute personne ou organisation la capacité d'écrire, de rédiger, publier, découvrir, obtenir, scruter, prioriser et, avec l'accord des parties prenantes directes, automatiser des règles sur tout réseau numérique, avec précision, simplicité, échelle, rapidité et résilience, tout en respectant les prérogatives, les accords et les préférences.

En dehors du domaine universitaire, ce document remplit une fonction simultanée de spécification versionnée pour les constructeurs de systèmes bêta de trois mises en œuvres de référence fonctionnelles. Il sert également de manuel de planification des opérations pour les équipes d'organisations des secteurs public et privé qui envisagent activement le déploiement et qui ont besoin d'une compréhension de la part de la direction sur la manière dont le système est conçu pour fonctionner.

1.2 La problématique

Une équipe de gestion de projet est chargée de prendre en compte les règles exprimées dans les contrats, les accords, la législation, les règlements, la jurisprudence, les avis, les directives, les normes, les manuels, les protocoles, les principes, les lignes directrices et les conventions informelles.

Les gestionnaires de projet ont besoin de pouvoir diffuser, découvrir et obtenir facilement les règles qui sont « en vigueur » pour les dates/heures données et les prérogatives relatives aux identités et aux juridictions qui les concernent ; qui sont « applicables » à la catégorie d'efforts et de tâches qu'ils entreprennent ; et qui sont « invoquées » par les circonstances particulières du moment.

Le problème a jusqu'à présent été négligé dans la littérature formelle de cette discipline, bien qu'il reçoive de l'attention dans l'industrie parce que l'inefficacité et les mauvaises performances constituent des opportunités entrepreneuriales pour les entreprises créatives de concevoir et de mettre sur le marché des méthodes de contournement, des solutions partielles et des solutions complètes (Dean & McMullen, 2002) (Driouchi & Bennett, 2012). Par exemple, un sous-secteur industriel entier d'entreprises concurrentes est apparu dans le but d'automatiser les taxes sur les transactions, les exemptions, les crédits et les droits d'importation/exportation, même si toutes les autorités fiscales du secteur public concernées auraient pu mettre en œuvre ces règles de manière générique et automatisée dès le départ.

Les services commerciaux d'automatisation peuvent être utiles pour aider les organisations, au niveau micro, à réduire l'inefficacité de la conformité et à réduire l'inefficacité due à l'échec de la conformité. Mais au niveau méso, c’est à dire, les systèmes de règles, cet arrangement ad hoc semble sous-optimal.

Cette recherche de conception fournit un raisonnement, une spécification fonctionnelle et des composants fonctionnels pour résoudre la catégorie générale de problèmes suivante :

L'agent A, qui interagit avec l'agent B, a besoin de connaître une ou plusieurs règles gérées de manière externe par les agents C..n qui sont « en vigueur » pour un contexte général, qui sont « applicables » à un ensemble de catégories d'événements et qui sont « invoquées » par des circonstances particulières, où :

(i) A et B peuvent ou non connaître les règles de C..n, ou leurs mises à jour, mais l'un ou l'autre ou les deux préféreraient obtenir tous les faits disponibles sur les règles pertinentes lors de l'interaction.

(ii) C..n peut ou non connaître A et B en particulier, ni leur moyen d'interaction particulier, mais peut s'attendre à ce que A ou B ou leur moyen d'interaction soit capable d'échanger des données avec un moyen générique commun à A..n.

(iii) A et B toléreraient le risque d'exposer des données limitées par le biais du support générique afin qu'elles puissent être utilisées pour sélectionner des informations sur les règles pertinentes de C..n.

Chapitre 2 : Méthodologie

2.1 Objectif et méthodologie d'un DBA par rapport à un PhD

Un DBA (doctorat en administration des affaires) est différent d'un PhD (doctorat en philosophie) dans le sens où, en général, un candidat PhD poursuit une recherche originale de valeur académique pour faire avancer la compréhension théorique, alors qu'un candidat DBA applique la théorie disponible pour résoudre une catégorie générale de problèmes du monde réel dans la pratique professionnelle.

2.2 Méthodologie de la théorie du milieu de gamme

Une carte conceptuelle permettant de comprendre la « théorie du milieu » est reproduite à la Figure 5, tirée d'un article de William Kuechler et Vijay Vaisshnavi, intitulé : « A Framework for Theory Development in Design Science Research » (Kuechler & Vaishnavi, 2012). Leur travail découle de la théorie de la conception des systèmes d'information (ISDT), qui est directement adaptée à la présente recherche de thèse, mais semble également facilement généralisable à tout domaine. Ce cadre est ensuite élaboré dans leur livre par le premier éditeur technique, CRC Press : « Design Science Research Methods and Patterns : Innovating Information and Communication Technology » (Vaishnavi & Kuechler, 2015).

Une TISD [théorie de conception des systèmes d'information] est, par sa nature et son intention, prescriptive. Une TISD est similaire à ce que l'on appelle un modèle en informatique et dans certaines disciplines de l'ingénierie [...]. [I]l fournit une définition de haut niveau du fonctionnement d'un artefact pour atteindre un objectif de conception et une direction vers sa construction, mais ne décrit pas comment l'artefact fonctionne ou par quel(s) mécanisme(s) les méta-exigences et la méthode de conception atteignent l'objectif de conception. La philosophe de la méthodologie de recherche, Nancy

Cartwright, a souligné récemment : La théorie de portée moyenne concerne les choses qui sont classées sous l'étiquette « mécanisme » (Cartwright, 2020, p. 269).

En plus de cette méthodologie limitée par les TISD, le présent travail est entrepris en parallèle avec des implémentations partielles testables entre professionnels indépendants, un type d'ingénierie simultanée de facto (Maranzana et al., 2008). La collaboration par le biais de licences et de relations libres/libres/ouvertes constitue un « examen par les pairs » indépendant qui est pertinent pour une thèse pragmatique de DBA orientée vers la résolution de problèmes du monde réel.

En dépit de ce qui vient d'être dit, cette thèse ne contient que très peu d'exemples de cas d'utilisation industriels ou gouvernementaux, limités à ceux qui permettent d'illustrer les éléments de conception du système. Cette contrainte s'explique par trois raisons :

(a) Les détails des projets particuliers du « monde réel » basés sur la conception nouvellement créée sont hors de portée de cette thèse de recherche sur la conception afin de maintenir l'accent sur le caractère général de la nouvelle conception DWDS.

(b) La plupart des véritables projets de mise en œuvre et de collaboration qui ont découlé de ma conception du DWDS sont organisés et dirigés par des personnes et des organisations indépendantes.

(c) Certains projets de mise en œuvre authentiques que je dirige personnellement sont mentionnés brièvement mais restent en dehors de la portée organisationnelle de cette thèse de recherche afin d'éviter tout conflit d'intérêt réel, apparent ou potentiel.

2.3 Recherche sur la conception

Cette recherche comporte des aspects qualitatifs et quantitatifs, non pas comme des entreprises distinctes en parallèle, mais comme des caractéristiques simultanées d'une méthode unifiée.

Quatre voies ont été empruntées au cours de cette recherche :

- Investigation et étude intensives de sources primaires et secondaires, académiques, industrielles et historiques dans une gamme étendue de domaines complémentaires.
- Partage proactif et discussion avec d'autres de mes recherches en cours par le moyen de licences et de forums libres/libres/ouverts, d'une application de gestion de projet en ligne, de médias sociaux et de conférences.
- Une collaboration structurée par le canal de la fondation à but non lucratif Xalgorithms, que j'ai créée pour gérer le financement de la recherche, établir des contrats de soutien et accueillir des groupes de travail.
- Réflexions sur mes propres engagements parallèles dans divers projets impliquant l'industrie, le gouvernement et d'autres organisations.

Ce projet est structuré dans la tradition de la « science de la conception » que Vijay Vaishnavi et William Kuechler décrivent comme « une recherche qui utilise la conception et la construction d'artefacts (apprentissage par la construction) pour générer de nouvelles connaissances et de nouveaux aperçus sur une classe de problèmes » (Vaishnavi & Kuechler, 2015, p. 396) (Chakrabarti & Lindemann, 2015) (Wieringa, 2014) Notre processus est décrit par Rudolf Sinkovics et Eva Alfoldi comme une « focalisation progressive non linéaire » (Sinkovics & Alfoldi, 2012) Il est réalisé sous forme de cycles itératifs multiples dans la méthodologie de « recherche inclusive de conception » d'Imre Horváth (Horváth, 2007) :

(a) Observer, décrire et réfléchir à l'état actuel des connaissances et des applications ;

(b) Inventer des concepts, des modèles, des méthodologies et des conceptions, fondés sur la théorie et la pratique ;

(c) évaluer la validité et la faisabilité des conceptions en construisant et en vérifiant des instanciations réelles.

Ce projet est également façonné par un style particulier de recherche en conception :

(d) Identifier et s'aligner sur les vertus et les normes de conception élémentaires, de la manière soulignée par Tim Berners-Lee dans « *Principles of Design* » (Berners-Lee, 1998a) ;

(e) Mettre en œuvre des prototypes fonctionnels dans la tradition pragmatique de la communauté des développeurs Internet (Russell, 2006), qui reflète « l'ingénierie simultanée » telle que définie par Nicolas Maranzana et Emmanuel Caillaud (Maranzana et al., 2008) et les « consortiums d'apprentissage » expliqués dans les travaux d'Edgar Schein (Schein, 1995) ;

(f) Poursuivre la « recherche engagée » en reliant le savoir-faire des praticiens de l'industrie aux travaux des théoriciens du concept, en suivant les conseils d'Andrew Van de Ven et de Paul Johnson (Van De Ven & Johnson, 2006).

(g) Fouiller les sources profondes de l'évolution des connaissances, comme l'encourage Popper (1979, pp. 238-239), ce qui peut nous amener à faire resurgir des idées et des techniques utiles qui ont été négligées ou oubliées.

2.4 Critères de réussite du projet

Le résultat visé par ce travail est de disposer d'un raisonnement, d'une spécification fonctionnelle et d'un prototype partiel de composants fonctionnels pour résoudre la classe générale de problèmes suivante : « *L'agent A, qui interagit avec l'agent B, a besoin de connaître une ou plusieurs règles gérées de l'extérieur par les agents C..n, qui sont en vigueur dans des contextes donnés, qui sont applicables à un ensemble de catégories d'événements et qui sont invoquées par des circonstances particulières dans un contexte d'incertitude sur les agents et les règles.* »

Le succès de cette recherche en conception peut être évalué en fonction de trois critères :

> *Est-il plausible que le système cible, une fois implémenté, soit capable de permettre à quiconque d'exprimer, de publier, de trouver et de récupérer des règles :*
>
> *1. À l'échelle sur l'Internet ? (faisabilité)*
>
> *2. Dans tous les domaines de règles et cas d'utilisation ? (généralisabilité)*
>
> *3. De manière plus efficace (meilleurs résultats) et/ou plus efficiente (moins de temps/argent/risque) ? (utilité)*

Ces critères ne peuvent pas être appliqués directement à un résultat cible de niveau méso, car cela nécessiterait une mise en œuvre ou un déploiement externe complet du service de bout en bout. Même si cette recherche conceptuelle est motivée par une volonté générale d'améliorer les règles et les pratiques dans des secteurs et des marchés entiers, les critères ne peuvent être appliqués qu'au raisonnement, à la spécification fonctionnelle et à la mise en œuvre partielle des composants fonctionnels.

La construction et les test partiels de nouveaux éléments du système cible, entrepris de manière itérative dans le cadre du processus de recherche en conception, aident le concepteur et les contributeurs participants à réfléchir à la cohérence de chaque partie et à la composabilité de bout en bout de l'ensemble.

Nonobstant ces limites de validation, en fait, tout au long de la première moitié de l'année 2022, une quantité considérable de travaux externes ont été entrepris pour créer une mise en œuvre de référence de bout en bout de la « version 3.x ». Ces travaux ont été menés de manière indépendante par des participants de la communauté free/libre et sur les termes *pro bono*, par des practiciens qui travaillent par ailleurs dans des organisations du secteur commercial et public.

La question pertinente est en fait la suivante : La méthode nouvellement conçue peut-elle être appliquée à l'échelle réelle ? Seule une initiative semi-indépendante à l'échelle permettrait de valider véritablement la nouvelle méthode d'expression. Actuellement, une petite sélection d'échantillons de règles testées fournit une démonstration raisonnablement crédible.

2.5 Vertus et normes de conception

Un ensemble explicite de vertus de conception fondées sur l'éthique et de normes de conception fondées sur le pragmatisme ont guidé notre recherche de conception de systèmes vers la spécification de données avec une directionnalité, et ont permis une cohérence de l'objectif tout au long de la trajectoire du projet.

2.5.1 Vertus de conception

La conception centrée sur l'humain, dans le domaine général des systèmes de règles informatiques, n'automatisera jamais l'imposition ou l'application impérative des règles. Au contraire, chaque personne soumise à une règle conserve en fin de compte sa prérogative inaliénable de discrétion quant à savoir si elle doit ou non, et dans quelle mesure, agir conformément à cette règle (Conrad, 1988). Sans une telle prémisse d'agence, la prérogative serait superflue.

La voie « libre/ouverte » combine le respect actif des libertés des utilisateurs avec la productivité des méthodes à logiciel libre. Il s'agit de perspectives distinctes mais complémentaires, du côté de la demande et du côté de l'offre, sur les relations entre les concepteurs, les opérateurs et les utilisateurs d'un système.

La tolérance englobe l'esprit de respect d'un concepteur à l'égard des prérogatives de ceux qui sont les utilisateurs d'une conception, ou qui sont soumis à son résultat, et d'autres concepteurs qui s'engageraient dans l'œuvre conçue en fonction de leurs modes, objectifs, domaines, technologies et paradigmes normatifs préférés.

L'interopérabilité est essentielle pour l'accès au marché et le choix de l'utilisateur. Lorsque les nœuds d'un réseau sont exploités sans lien de dépendance par des organisations concurrentielles autonomes, ils négocient des spécifications, des protocoles et des conceptions de composants communs par le cadre d'une gouvernance coopérative.

2.5.2 Normes de conception

La simplicité a été le mieux exprimée par Antoine de Saint-Exupéry dans sa façon de concevoir un avion : « En toute chose, la perfection est finalement atteinte non pas lorsqu'il n'y a plus rien à ajouter, mais lorsqu'il n'y a plus rien à retrancher. » (Saint-Exupéry, 1939, pp. 41-42) Doug McIlroy, l'un des principaux contributeurs à Unix et à la philosophie d'Unix, a souligné que la qualité, la lisibilité et la vitesse du code source d'un programme informatique sont améliorées en diminuant le nombre de lignes de code. Il a fait une déclaration célèbre : « Le véritable héros de la programmation est celui qui écrit du code négatif ». (McIlroy, 2009)

La modularité signifie que chaque élément d'un système sera interchangeable avec d'autres implémentations indépendantes. Cela permet de concentrer les spécifications d'un système sur les fonctionnalités de base, tout en étant libéré des détails de mise en œuvre qui ne sont pas intrinsèques aux spécifications fonctionnelles, comme la plate-forme, le langage, l'infrastructure, les services, le fournisseur et d'autres facteurs.

L'intuitivité est une qualité tellement essentielle qu'elle en devient inintéressante. Christopher Alexander a expliqué que les idées de conception les plus essentielles sont généralement les plus difficiles à percevoir « parce qu'elles sont si ordinaires qu'elles frappent au cœur ». Il observe que : « Ce qui les rend difficiles à trouver n'est pas qu'elles soient inhabituelles, étranges ou difficiles à exprimer - mais au contraire qu'elles soient si banales. » (Alexander, 1979, p. 219)

La décentralisation en droit (de jure) et/ou en pratique (de facto) signifie que le pouvoir à tout niveau d'agrégation est délégué à la discrétion des décideurs relativement désagrégés ou au niveau micro. En droit, on appelle cela la « subsidiarité ». C'est le contraire de la centralisation, dans laquelle le pouvoir à tout niveau d'agrégation est délégué à la discrétion des décideurs relativement plus polyvalents. En droit, cela s'appelle la « primauté ».

La moindre puissance consiste à choisir le langage le moins expressif possible pour un objectif donné, afin d'optimiser l'utilité des données. (Berners-Lee & Mendelsohn, 2006) Cela réduit le travail de computation nécessaire pour traiter les données (Bush & Meyer, 2002). renforce la sécurité informatique intrinsèque (Qureshi, 2017), est plus rapide à exécuter, et plus facile à comprendre et à retenir pour les gens (Zhuge 2010, p. 202).

Le style déclaratif tabulaire dans la programmation informatique présente les données d'entrée et de sortie dans des listes ordonnées simples (tuples). Il peut être utilisé pour les processus de mise en correspondance et de tri des données, qui sont beaucoup plus rapides à traiter que si l'on essayait d'exécuter les mêmes fonctions avec des commandes conditionnelles qui doivent être analysées et traitées dans une séquence étape par étape (Cunneyworth, 1994).

Chapitre 3 : Un examen réflexif de la littérature sur la nature d'une « règle »

3.1 Qu'est-ce qu'une règle ?

Une règle est une relation directionnelle communiquée entre deux ou plusieurs personnes pour associer ce qui « est » et ce qui « devrait être ». On ne peut pas déduire « devrait être » de « est », comme le soutenait David Hume il y a trois siècles (Hume, 1738). Mais tout le monde peut affirmer une direction de « est » vers « devrait être », que nous représentons ici par l'expression formelle : SOIT + REGLE ⟹ DOIVE. En matière pratique, logique, éthique et esthétique, les règles expriment l'obligation, la permission ou l'encouragement par le biais des termes couramment utilisés en majuscules DOIT, PEUT et DEVRAIT, ou de leurs différents négatifs et synonymes. (ISO/IEC, 2018). (Bradner, 1997).

L'expression formellement normalisée des règles dans les documents officiels de l'industrie, du commerce et de la gouvernance est apparue en même temps que les fondements philosophiques modernes de la théorie normative dans les années 1950 et 1960 (Wittgenstein, 1953) (Kripke, 1982) (Bloor, 1997)(Baker & Hacker, 2009). Notre recherche conceptuelle est guidée en particulier par les « Investigations philosophiques » de Ludwig von Wittgenstein (Wittgenstein, 1953), telles qu'elles sont élaborées dans la « Logique déontique » de Georg Henrik von Wright (Von Wright, 1951), la « Théorie des propositions normatives » de Jerzy (Georges) Kalinowski (Kalinowski, 1953) et l'ouvrage de Gertrude Anscombe, « Intention » (Anscombe, 1957). Ces philosophes ont distingué les modes d'expression impératif, déclaratif et empirique des règles qui sont « en vigueur » pour un contexte général, qui sont « applicables à une classe de circonstances » et qui sont « invoquées » par des événements particuliers :

Une règle est en soi une déclaration impérative d'obligation, d'option ou d'attente entre des personnes ;

La documentation relative à une règle (une « proposition normative ») est une déclaration de fait ;

L'applicabilité d'une règle à un événement particulier est une déclaration empirique de déduction.

Ces trois catégories sont souvent confondues dans la littérature publiée, car les techniques employées pour résoudre et communiquer tous ces problèmes logiques sont les mêmes.

3.2 Qu'est-ce que l'agence ?

Nous utilisons le terme « agence » pour indiquer la possession de la faculté d'action attitudinale, intellectuelle et tangible de poursuivre un résultat spécifique. Par exemple, le « gestionnaire de projet » a été défini pour la première fois en termes similaires par Paul Gaddis comme une personne souhaitant atteindre un résultat planifié « à temps, dans les limites du budget et conformément à des spécifications de performance prédéterminées » (Gaddis, 1959).

La prérogative sociale d'établir des règles peut impliquer des relations de subsidiarité fondées sur le pouvoir discrétionnaire des micro-décideurs relativement désagrégés, ou de prépondérance fondée sur le pouvoir discrétionnaire des décideurs relativement plus agrégés ou de niveau macro. Nous pouvons alors utiliser le terme « méta-règle » pour désigner une règle qui rend explicite cette répartition des prérogatives. Ainsi, ce serait une méta-règle qui établirait si l'agence d'un gestionnaire d'algorithme doit être prioritaire par rapport à celle d'un gestionnaire des opérations, ou vice versa.

On peut supposer qu'il faut se conformer aux règles légitimes. Pourtant, le fait de ne pas se conformer à une règle applicable dans un scénario particulier ne signifie pas nécessairement qu'on la viole. Un pouvoir discrétionnaire débridé risque d'entraîner un comportement impulsif ou opportuniste, tandis qu'une pression de conformité trop forte risque d'inculquer des codes de comportement trop rigides (Espedal, 2007). Pour permettre la discrétion dans l'évaluation d'une règle par rapport à toute autre règle, l'enveloppe de métadonnées délivrant une règle donnée pourrait inclure les trois variables contextuelles suggérées par trois spécialistes universitaires de la linguistique, An Verhulst, Ilse Depraetere et Liesbet Heyvaert (Verhulst et al., 2013) :

- *Source* : autorité de jure et/ou origines de facto d'un règlement, d'une condition ou d'une circonstance cible ;
- *Subjectivité* : l'implication ou l'engagement des bénéficiaires en vue de garantir le respect de la réglementation ;
- *Force* : la gravité ou l'impossibilité de la non-conformité.

3.3 Qu'est-ce qu'un algorithme ?

Le terme algorithme est dérivé du nom d'un érudit perse du 9ème siècle, Abi Jacfar Muhammad ibn Miisa al-Khwiirizmi. (Crossley & Henry, 1990) Il s'agit de tout ensemble d'opérations invoquées par une condition, afin d'exécuter une procédure pour résoudre un problème général et bien spécifié, qui se termine une fois la procédure exécutée. (Skiena, 2008, p. 3)

Un algorithme est une extension précise et composable de l'agence humaine. Sven Nyholm parle d' « agence collaborative », où les gens partagent l'agence avec des machines informatisées (Nyholm, 2018), et Douglas Rushkoff parle d'abandonner notre agence aux machines (Rushkoff, 2010, p. 14).

Cependant, il semble que ce soit une erreur logique de dépeindre quelque chose conçu et entretenu par des agents comme possédant lui-même une agence. Certes, une machine dotée de capacités d'automatisation avancées peut donner une impression d'agence. Pourtant, elle est sous le contrôle de ses concepteurs et de ses programmeurs. Une analogie partielle est un personnage de film qui peut donner l'impression d'être, de faire ou de ressentir, mais le public sait que c'est en réalité le scénariste, le réalisateur, l'acteur et l'équipe de production qui déterminent le personnage.

La notion bien établie d' « interaction homme-machine » confond la conception de l'interface utilisateur et l'agence. Dans une lutte à la corde, le contact physique se fait entre vos mains et la corde, mais la lutte n'est pas entre vous et la corde ; elle est entre vous et la personne à l'autre bout de la corde. Les humains interagissent avec d'autres humains par le biais de signaux radio, d'images, de programmes informatiques et d'algorithmes.

L'algorithme informatique doit être compris comme une extension évolutive précise et composable de l'agence humaine. L'agence est la possession d'une faculté d'action attitudinale, intellectuelle et tangible pour poursuivre un résultat spécifique. Un algorithme est une méthode invoquée par une condition pour obtenir un résultat spécifié, puis se terminer. Mais de quelle agence s'agit-il : celle de l'utilisateur de l'algorithme ou celle du concepteur de l'algorithme ?

Kees van Dongen et Peter-Paul van Maanen recommandent les pratiques suivantes pour s'assurer que les gestionnaires d'algorithmes soutiennent adéquatement les gestionnaires d'opérations qui s'appuient sur eux (van Dongen & van Maanen, 2013) :

Comprendre

- Utiliser des algorithmes modulaires simples pour aider les décisions
- S'assurer que les opérateurs connaissent chaque aide à la décision
- Rendre le raisonnement de chaque aide à la décision facilement disponible et compréhensible.
- Révéler les résultats intermédiaires de manière compréhensible.

Responsabilisation

- S'assurer que les personnes se sentent responsables de la performance
- Veillez à ce que les personnes soient tenues responsables de la qualité du résultat.

Gestion des erreurs

- Rendre transparentes les sources potentielles d'erreurs et les erreurs réelles.
- Aidez les opérateurs à prendre conscience des biais du système et de leurs propres biais.
- Informez les opérateurs des conditions dans lesquelles l'aide à la décision fonctionne bien et de celles dans lesquelles elle ne fonctionne pas.

3.4 Systèmes de transmission des règles

3.4.1 Signal et bruit dans les systèmes de transmission de règles : aperçu de la théorie de l'information

Dans son article de 1948, Claude Shannon a proposé un moyen pour mesurer la quantité d'information dans un message numérique. Son modèle général de communication relie la source et la destination de l'information, avec une source de bruit (Shannon, 1948). Dans notre contexte, une règle est communiquée comme une proposition normative dans un message, transmis comme un signal avec une certaine quantité de bruit entre la transmission et la réception.

Nous pouvons supposer que la source de l'information est toute personne ayant une prérogative sociale en tant qu'« agent générateur de règles », et que la destination de l'information est tout « agent générateur de règles » considéré comme assujetti à ces règles sur la base de l'autorité, de l'accord ou de la préférence. La règle elle-même est une obligation, une permission ou un encouragement connu dans un contexte social entre individus et entités. Ce n'est donc pas la règle impérative en soi qui constitue la charge utile transmise, mais un avis déclaratif sur une règle. Le concepteur d'un système de règles facilite la communication des propositions normatives.

3.4.2 Trois postulats pour des systèmes des règles, ensembles de règles et règles optimales

Le résultat de cette recherche sur la conception de « meilleurs » systèmes de règles devrait être de permettre aux individus et aux entités de communiquer des propositions normatives de manière plus coût-efficace et plus coût-effective que ce qui est actuellement faisable :

- *Le rapport coût-efficacité* : Maximiser la qualité de la communication sur les données qui possédent ~~de~~ direction dans un délai, des ressources et des risques entendus.
- *Le rapport coût-effectivité :* Minimiser du temps, des ressources et des risques nécessaires pour atteindre une certaine qualité de communication des données qui possédent ~~de~~ direction.

Ces critères de coût-efficacité et de coût-effectivité se retrouvent de manière à peu près équivalente dans les quatre principales normes internationales de compétences des chefs de projet. (ISO, 2012, Section 4.3) (IPMA, 2006, Section 3.10, p 150-153) (Crawford & Duncan, 2007, p. 4) (ICCPM, 2012, Views 3, 6, 9).

Les objectifs de coût-efficacité et de coût-effectivité peuvent sembler être des critères praticables en soi, mais ils dépendent tous deux d'une notion indéterminée de « qualité ». C'est pourquoi nous les étendons avec trois postulats pour la qualité de la conception des systèmes de règles :

- *1. Un système de règles optimal au sein d'un groupe juridictionnel d'individus et d'entités arbitraires est celui qui leur demande le moins d'efforts pour catégoriser et communiquer leurs propositions normatives respectives.*
- *2. Une règle optimale entre deux individus ou entités quelconques est celle qui est centrée sur leurs priorités respectives, tout en recoupant leurs points d'accord communs.*
- *3. Un ensemble optimal de règles entre plusieurs individus ou entités est celui dont l'ensemble des règles révèle un raisonnement émergent bien aligné.*

3.4.3 Considération des systèmes de transmission des règles à la lumière de l'écologie théorique

Cette recherche positionne les systèmes de règles au niveau « méso » d'un cadre « micro-meso-macro » en relation avec une théorie de l'agence de type « écologie de projet ».

Kurt Dopfer, John Foster et Jason Potts fournissent un cadre général pour conceptualiser les systèmes de règles en tant que cadres fonctionnels des projets. Ils fournissent une interprétation du cadre à trois niveaux de Joseph Schumpeter qui distingue le micro (comportements des décideurs parmi les organisations), le méso (règles, infrastructures normatives et pratiques parmi les industries et les marchés) et le macro (caractéristiques de sociétés et d'économies entières) (Dopfer, 2012) (Dopfer et al., 2004) (Schumpeter & Boody, 1954). Ils proposent que les efforts pour obtenir des améliorations de performance à travers le niveau micro, où les gestionnaires de projet opèrent, peuvent être poursuivis plus efficacement au niveau méso où l'infrastructure normative et les pratiques des industries et des marchés sont formulées et gérées. Un changement au niveau méso qui est massivement adopté au niveau micro peut avoir des effets émergents et transformateurs au niveau macro pour des sociétés et des économies entières. La question de savoir si ces résultats sont « meilleurs que », « équivalents à » ou « pires que » le scénario précédent ou les résultats alternatifs repose sur une prémisse concernant la voie à suivre.

À peu près à la même période, le géographe économique autrichien Gernot Grabher a introduit l'expression « écologie de projet » dans la littérature sur la gestion de projet en 2004 pour tenir compte de « l'interface entre les projets et les organisations, les communautés et les réseaux dans et à travers lesquels les projets fonctionnent » (Grabher, 2004). Ce cadre micro-méso-macro est recommandé dans l'Oxford Handbook of Project Management (Grabher & Ibert, 2011) (Grabher, 2004), et par l'un des fondateurs du domaine de la gestion de projet, Jonas Söderlund (Söderlund, 2004). Elle est également présentée dans deux numéros spéciaux de l'Academy of Management Review (Bies et al., 2007) (Hitt et al., 2007). L'écologie de projet encadre un projet à plusieurs niveaux : « du micro-niveau des réseaux interpersonnels au méso-niveau de la collaboration intra- et inter-organisationnelle, en passant par le macro-niveau des cadres institutionnels plus larges ».

Dans une revue de la littérature publiée il y a dix ans, Söderlund a observé que les études de projets multiples impliquant des environnements et des règles institutionnels auxquels l'organisation individuelle doit répondre et se conformer, n'avaient pas reçu une attention suffisante dans les principales revues universitaires de gestion de projet (Söderlund, 2004, p. 483).

Les projets de niveau méso visant à concevoir et à mettre en œuvre des plateformes de réseau nouvelles ou améliorées peuvent affecter des industries, des marchés et des économies entières. Les changements de niveau méso peuvent étendre ou éroder la capacité et l'agence corporelle et intellectuelle de l'homme de la manière décrite par Ernst Kapp, Alfred Lotka, Nicholas Georgescu-Roegen et Karl Popper (Kapp, 1877) (Lotka, 1925) (Georgescu-Roegen, 1975) (Popper, 1979). Lorsque de nouveaux systèmes de niveau méso améliorent ou dégradent les performances tangibles des individus ou des organisations au niveau micro, il en résulte des changements comportementaux progressifs qui peuvent finalement se matérialiser par des effets émergents et transformateurs au niveau macro pour des sociétés et des économies entières. Compte tenu de cette séquence méso-micro-macro, un projet de mise en œuvre d'une plate-forme réseau modifiée, bien que directement instancié par des modifications des opérations quotidiennes, peut en fait être conçu pour modifier indirectement des industries, des marchés et des économies entiers.

Chapitre 4 : Examen de la recherche sur des techniques antérieures. concernant la conception de systèmes de règles

4.1 Des techniques antérieures pour l'expression de la logique des règles

Une règle communiquée en langage naturel peut également être exprimée et considérée sous diverses autres formes, comme un langage de programmation informatique, un organigramme graphique (diagramme en arbre) et une table d'entrée/sortie du type « table logique », « table de vérité », « table de contrôle » ou « table de décision ». Dans chaque cas, il s'agit de comparer un ensemble de données d'entrée avec un ensemble de conditions, d'hypothèses, de déclencheurs ou de prémisses, et les données de sortie sont générées sous forme d'instructions, de réponses, d'actions, d'options ou de conclusions.

Dans les sections suivantes, nous procédons à une comparaison directe de différentes manières d'exprimer la politique de livraison d'une épicerie.

La forme la plus simple de règles en tant que prose semble être RuleSpeak, qui réduit les règles à des énoncés déclaratifs concis, cohérents et non ambigus (Ross, 2009). Cependant, RuleSpeak n'est pas un langage, un schéma ou une syntaxe. Il s'agit d'un guide incorporant un ensemble de « meilleures pratiques » conformes à la norme SBVR (Semantics of Business Vocabulary and Business Rules) de l'OMG (OMG, 2016b). L'avantage de RuleSpeak est que son utilisation formelle du langage naturel pour la modélisation descriptive ou les règles et la logique peut être facilement comprise par les humains et traitée par la machine tant que le système dispose d'un analyseur syntaxique approprié (Chapin, 2008). Toute personne connaissant l'anglais peut lire et suivre ces trois affirmations. En revanche, un logiciel sophistiqué de traitement du langage naturel serait nécessaire pour analyser l'une d'entre elles et permettre à la machine de déterminer quel résultat est invoqué en fonction des circonstances.

La même règle peut être exprimée en code de programmation informatique pour la détermination par la machine : Ruby de programmation procédurale, RuleML qui est déclarative et Notation3 déclarative. Cependant, ces exemples de règles en tant que code contiennent des styles d'expression et de balisage qui sont propres à chaque langage de programmation ou de balisage. Ces styles uniques font que, lorsque l'une de ces expressions de règles en code est partagée, les destinataires ont besoin d'une application logicielle capable d'analyser les expressions et le balisage spécifiques au langage. Sinon, les destinataires devraient les remanier ou les réécrire pour qu'elles fonctionnent sur une plate-forme d'application différente.

4.2 Des techniques antérieurs pour les modèles de données logiques

Les éléments de « table de vérité » couramment utilisés {F, T, - | X} représentent une des nombreuses variantes de l'expression tabulaire des relations logiques. Cette section rend compte d'un examen d'autres modèles de données logiques qui a été entrepris afin d'envisager des idées pour une approche qui sera significative pour la logique normative générale, et qui sera également utilisable par les humains et les machines.

Les concepteurs de systèmes informatiques ont principalement représenté l'information logique avec des bits ayant deux états {0,1}, en commençant par l'introduction de la logique binaire électronique à ce but à la fin des années 1930 par John Atanasoff et Clifford Berry (Gustafson, 2000). Le choix entre des chiffres ou des lettres peut sembler sans conséquence, mais les portes logiques avec {0,1} ou {-1, 0, 1} sont précises, alors que les systèmes qui emploient {T,F} présentent une ambiguïté intrinsèque. Les termes « vrai » et « faux » peuvent être interprétés de diverses manières, et les « tables de vérité » ont des origines, des styles et des significations multiples dans des contextes particuliers (Anellis, 2012) (Shosky, 1997). La signification du troisième symbole diffère dans chacun des systèmes logiques trinaires, comme « ni l'un ni l'autre « {T,F,N} » et « n'a pas d'importance » dans {T,F,-}. Jan Łukasiewicz a introduit la logique « multivaluée » en 1920 (Łukasiewicz, 1920), qui a été affinée par Lotfi A. Zadeh sous le nom de « logique floue » (Zadeh, 1965). Ces systèmes utilisent le troisième élément comme une valeur proportionnelle entre 0 et 1. Le troisième élément est essentiel à l'informatique quantique {0, ψ, 1}, où le symbole grec Psi 'ψ' est utilisé pour représenter une attente probabiliste contingente de 0 OU 1.

Pourtant, il existe toute une catégorie de problèmes qui ne peuvent être résolus avec davantage de données, de connaissances ou de temps. Il existe des paradoxes insolubles dans la théorie mathématique formelle des ensembles, exprimés succinctement avec l'exemple de Bertrand Russell : « Il existe une proposition que j'affirme et qui est fausse. » (Russell, 1919, p. 356) impliquant une superposition logique comme « vrai ET faux », « 0 ET 1 », ou « oui ET non ».

Il ne s'agit pas d'une digression abstraite. Des contradictions logiques apparaissent parfois dans le monde réel du droit, des affaires et de la politique. (Fletcher, 1985) (Cook, 1924) Ceci est clairement illustré par une décision de la Cour d'appel de Californie du 31 mai 2022, dans laquelle il a été déterminé qu'en tenant compte du but de la section 45 de la loi californienne sur les espèces menacées (Fish & G. Code), les bourdons sont des poissons. Un concepteur de systèmes appliqués pourrait considérer chaque contradiction comme une erreur, ou faire en sorte qu'un processus opérationnel soit suspendu jusqu'à ce qu'un humain résolve la contradiction. Mais le concepteur peut aussi reconnaître que la contradiction est parfois un véritable état persistant, et la traiter comme tel. Susan Sturm, professeur à la Columbia Law School, propose de « concevoir pour le paradoxe ».

Pour notre objectif particulier, nous empruntons certains concepts de logique « tetranaire » (c'est-à-dire quatre symboles, par contraste avec le « quaternaire », qui est composé de quatre numéraux) tels que {T,F,B,N} Vrai, Faux, Les deux, Ni l'un ni l'autre ; ainsi que {A,T,C,G} et {A,U,C,G} de la computation moléculaire.

4.3 Distinguer la conception actuelle pour le traitement de la logique des règles

Cette thèse présente et explique le fonctionnement d'un nouveau type de système de règles appelé « Data With Direction Specification » (DWDS, ou de données avec une directionnalité). Avant de détailler ce qu'est une chose, il peut être utile de préciser ce qu'elle n'est pas.

À divers égards, les fonctions du DWDS ressemblent à celles des moteurs de règles, des processus de flux de travail, des systèmes d'aide à la décision, des contrôleurs logiques programmables et de l'intelligence artificielle. Mais il diffère de chacune de ces niches de conception par des aspects essentiels.

4.3.1 Distinguer le DWDS des systèmes de contrôle logique programmable (PLC)

Le DWDS apporte aux réseaux distribués une sorte de porte logique à usage général qui ressemble au contrôleur logique programmable (PLC) de Richard Morley conçu dans les années 1960 qui fournissait la séquence suivante aux machines individuelles :

[données d'entrée] → [porte logique] → [données de sortie] (Brown, 2015) (Amin & Mridha, 2020).

L'automate requiert des classes connues d'agents, de scénarios et de séquences de règles qui peuvent être mises en œuvre dans des configurations individuelles, en série en cascade et multiplexées en parallèle. La programmation E/S de la logique de l'automate se fait dans un style impératif avec des données, une logique et une procédure étroitement intégrées.

Nous avons plutôt cherché à concevoir une spécification adaptée à l'incertitude des réseaux distribués ouverts et auto-organisés, composés de types d'agents illimités, de scénarios partiellement inconnus et de structures et séquences de règles définies de l'extérieur. Pour que ce type de contrôleur logique fonctionne, il est essentiel de dissocier les données, la logique et la procédure, et d'exprimer les tables logiques d'entrée-sortie (E/S) dans une forme déclarative tabulaire qui ne dépend pas de la plate-forme informatique.

- ENTRÉES / SORTIES :
  { non | oui | ouiETnon | ouiOUnon } / { w | x | y | z }

L'utilisation de cette structure pour la logique normative est l'une des diverses classes de cas d'utilisation intéressants. Sur le plan programmatique, les étiquettes qui apparaissent dans un paquet de données pourraient signaler dynamiquement à RR et RT de puiser dans un ensemble normalisé de sémantique parsible pour la partie inférieure du table verticale d'E/S. Sur le plan opérationnel, le DWDS décrit une porte logique polyvalente à valeurs multiples (Ebrahimi et al., 2016) pour le traitement des données sur tout type de réseau de calcul. Certaines options sont :

- CONDITIONS/AFFIRMATIONS :
  { non | oui | ouiETnon | ouiOUnon } / { pas | doit | peut | devrait }
- OBSERVATIONS/EXPECTATIVES :
  { non | oui | ouiETnon | ouiOUnon } / { w% | x% | y% | z% } quartiles de distribution
- SUPPOSITIONS/CONCLUSIONS :
  { non | oui | ouiETnon | ouiOUnon } / { déductif | inductif | hypothétique | transductif }

4.3.2 Distinction entre le DWDS et un moteur de règles

Le DWDS est similaire à un moteur de règles en ce sens que « le système parcourt toutes les règles, choisit celles pour lesquelles la condition est 'oui' , puis évalue les actions correspondantes » (Fowler, 2009).

Cependant, les plateformes moteur de règles sont généralement censées fonctionner avec une application centrale et une base de données de règles codées, le tout dans un langage expressif générique tel que Python, Lisp, JavaScript, Prolog ou un langage spécifique au domaine comme RuleML ou XBRL-Formula.

Notre conception représente un autre type de recherche : une spécification à usage général pour communiquer des règles en tant que données, d'une manière qui est utilisable de façon équivalente par n'importe quelle application, dans n'importe quel langage, sur n'importe quel dispositif, sans rétrofits ni remaniements. Aucun type particulier de « moteur de règles » n'est nécessaire pour traiter ces données, car les enregistrements conformes à la DWDS fonctionneront dans n'importe quel environnement. Un réseau RuleReserve décentralisé de type peer-to-peer permet à chaque nœud autonome d'accueillir dynamiquement n'importe quelle API des applications périphériques exploitées par les utilisateurs finaux.

4.3.3 Distinguer le DWDS d'un système d'aide à la décision.

On peut considérer qu'en général, les systèmes de décision automatisés sont basés sur « une série de paires de conditions IF-THEN-ELSE » (Bidgoli, 2015, p. 268) dans lesquelles un état futur probabiliste contingent (IF) entraîne des exigences déclaratives conséquentes (THEN) (Mladenic et al., 2012, p. 8-9) (Keen & Scott Morton, 1978).

En revanche, un système basé sur le DWDS fonctionne avec des données générées à partir d'un état d'entrée empirique (GIVEN a,b,c et WHEN d,e,f), qui est utilisé pour passer au crible les exigences déclaratives conséquentes de sortie (THEN g,h,i), afin d'informer les utilisateurs (North, 2006) (Fowler, 2013). Le résultat est une simple assertion normative :

DONNANT des données contextuelles ;
QUAND des données particulières apparaissent également ;
ALORS certaines déclarations de sortie sont invoquées.

4.3.4 Distinguer le DWDS de l'intelligence artificielle

L'intelligence artificielle (IA) est largement définie par l'acquisition de connaissances par la machine et le raisonnement inductif complexe pour résoudre les lacunes d'information afin de guider l'action. La méthodologie de raisonnement la plus courante à laquelle il est fait référence dans ce contexte est l'inférence variationnelle stochastique (Hoffman, 2013)(Plötz et al., 2018).

La portée du DWDS est uniquement de permettre la communication de propositions normatives, sans inférence et sans usurper les prérogatives d'autrui pour guider l'action. Nous appelons cela la Naïveté Artificielle et nous la représentons par le symbole du zéro barré, ∅A (*néant-ah*), qui désigne en mathématiques un ensemble vide, c'est-à-dire un ensemble comportant zéro éléments { }. Les questions relatives au contexte, à la compréhension, au but, à la raison, à l'apprentissage ou à la motivation sont toutes considérées comme des prérogatives de l'utilisateur final, à traiter à la périphérie du réseau. Dans cette conception, les opérateurs des nœuds RuleReserve ne fournissent rien d'autre que la structuration et le stockage des données, ainsi que des services de demande-réponse et de transmission en temps réel. Chaque nœud utilise les données passives des messages transitoires pour effectuer des opérations de filtrage (aussi appelées « correspondance des signaux ») sur les règles persistantes stockées.

Un nœud RuleReserve ne dispose d'aucune méthode pour conserver ou copier des données ou métadonnées relatives au contexte, à la compréhension, au but, à la raison, à l'apprentissage ou à la motivation. Il ne fait que renvoyer une réponse à la source de la demande d'exécution. Le demandeur est responsable de ses propres journaux d'activité vérifiables.

4.3.5 Distinction entre le DWDS et le déroulement des processus d'affaires

Un déroulement de processus d'entreprise est une séquence interactive par étapes de tâches (Gantt, 1919) (Geraldi, 2012) dans laquelle des tâches complexes, volumineuses ou simples et cohérentes peuvent être automatisées à l'aide de divers types de matériel et/ou de logiciel, tandis que des décisions sont prises en cours de route par des personnes.

Le DWDS maintient « l'agent créateur de règles », qui rédige les règles en entrée, à une distance fonctionnelle de « l'agent récepteur de règles », qui utilise les résultats directement ou par l'intermédiaire de sa machine. Il ne décrit qu'un « pipeline de traitement des données » qui fonctionne selon un déroulement sans interruption à travers une seule séquence d'entrée-sortie informatique (von Landesberger et al., 2017). Bien qu'il existe des états de sortie intermédiaires des données qui deviennent des entrées pour les étapes suivantes, il n'y a pas de séquence de décisions.

4.3.6 Distinguer le schéma de règles du DWDS d'un langage spécifique au domaine

Le DWDS introduit une nouvelle manière structurée d'exprimer, de communiquer et de rendre opérationnelles les règles en utilisant n'importe quel langage naturel. Cependant, il ne s'agit pas d'un « langage spécifique au domaine » (DSL). Il s'agit simplement d'une spécification syntaxique pour supporter une configuration de données de règles pour la programmation par tuples (Underwood, 2011). Dans nos mises en œuvres de référence, ces données sont emballées avec la syntaxe JSON (JavaScript Object Notation) (Bray, 2014), mais c'est un détail. Les mêmes données peuvent être exprimées de manière équivalente dans une autre implémentation en utilisant CBOR (Concise Binary Object Representation, une représentation de données binaires vaguement basée sur JSON) (Bormann & Hoffman, 2013), ou CSV (comma separated value), ou comme des champs de base de données séparés sans aucune syntaxe associée.

En séparant chaque phrase de règle en un ensemble réutilisable de six éléments syntaxiques, et en séparant également ceux-ci d'un tableau contenant les relations entre les conditions et les assertions, notre méthode accepte la séparation recommandée par Robert Kowalski de la logique et du contrôle. (Kowalski, 1979, 435)

Le DWDS n'impose aucune exigence de programmation au-delà de la structure syntaxique à six éléments. Le DWDS n'impose pas le choix d'un langage de programmation particulier. De plus, le DWDS permet aux auteurs de règles d'écrire et de lire les règles dans la langue de leur choix, avec ou sans termes ou style spécifiques au domaine. Le choix du style d'expression est une question de jugement de l'utilisateur final.

Le DWDS crée cependant une incitation à utiliser des schémas et des lexiques communs, une approche qui permet d'éviter la tendance à la redondance et à l'incohérence qui est apparue parmi les schémas standard concurrents (Sliwa & King, 2000). Nous avons conçu une incitation pratique pour que l'alignement sémantique émerge par le biais de la coopétition (Brandenburger & Nalebuff, 1997), mais qui est laissé à son émergence propre, indépendamment de la spécification en soi. L'incitation est suffisante.

4.4 Influences de techniques antérieures et inspirations de 70 ans de logique programmable

Cette section offre un aperçu des concepts et méthodes techniques qui ont façonné le parcours de cette recherche. Mes propres décisions de conception ont été éclairées par l'analyse de ces sources de « données primaires » sur la conception de la logique programmable (par exemple, sur les styles d'expression des règles, les méthodes de traitement des données). Il s'agit d'une reconnaissance générale des techniques antérieurs auquels j'ai explicitement réfléchi, et que j'ai utilisé directement ou indirectement. Elle comprend quelques références à des méthodes qui m'ont aidé à réfléchir à ce que je voulais faire différemment.

Cette section est présentée sous quatre thèmes considérés comme contextuellement pertinents pour ma propre perspective de recherche : Structuration et transmission des données ; Programmation logique tabulaire ; Programmation logique procédurale ; et Un modèle de données pour la logique tetranaire. Chaque thème est rassemblé par décennies.

La redécouverte et la réhabilitation d'antériorités oubliées depuis longtemps, datant des années 1950, 1960, 1970 et 1980, constituent une part importante de cette entreprise de recherche en conception. La plupart des entrées incluses ici sont bien connues, mais beaucoup d'entre elles ont nécessité une recherche méticuleuse des sources originales, afin d'assurer une reconnaissance appropriée. En particulier, la programmation déclarative tabulaire a nécessité un effort d'excavation déterminé.

Chapitre 5 : DWDS : raisonnement technique et résumé de la conception

5.1 Les données avec une directionnalité : Des concepts à la spécification d'un système

Nous proposons le descripteur « Data With Direction Specification » (DWDS; ou les données avec une directionnalité) pour décrire tout système distribué à usage général que des particuliers et des organisations peuvent utiliser pour rédiger, publier, découvrir, obtenir, scruter, prioriser et, avec l'accord des parties prenantes directes, automatiser les règles à travers des réseaux numériques avec précision, simplicité, échelle, vitesse, résilience et respect de la prérogative.

DWDS est une spécification pour une classe de pipeline de processus de données résumée par la relation fondamentale suivante .

$$\text{SOIT} + \text{REGLE} \Longrightarrow \text{DOIVE}$$

Le DWDS implique trois fonctions.

- *RuleMaker :* Un utilisateur possédant des prérogatives sociales ou institutionnelles pour l'élaboration de règles rédige l'expression normative, et maintient l'historique de ses versions, et l'ajoute dans un fichier de réserve d'accès général qui est diffusé sur Internet, et monitore l'activité directe pour la règle.
- *RuleReserve :* Toute personne qui s'organise pour mettre en place et gérer un nœud Superset RuleReserve peut obtenir le dernier registre complet [rulereserve.dwd] via sa diffusion permanente sur l'Internet ; et toute personne qui gère un nœud Subset RuleReserve peut effectuer une sélection dans cette collection distribuée. Les agents qui prennent des règles peuvent alors obtenir directement ou par l'intermédiaire de leurs machines, à la demande, des sélections de règles tamisées à partir de n'importe quel(s) nœud(s) qui leur est (sont) accessible(s) à travers des nœuds en cascade sur le réseau. Des mises à jour et des contrôles d'intégrité automatisés peuvent être exécutés sur les nœuds du Superset RR audités et certifiés.
- *RuleTaker :* Un agent utilisateur qui opère n'importe quelle application pour atteindre son but peut envoyer un message de demande contenant des données sur un événement procédural ou un changement d'état, afin de recevoir en retour un message de réponse contenant des données tamisées et vérifiées sur des propositions normatives et des données de contrôle correspondantes. Après avoir été informé de ce qui apparemment DOIT, PEUT et DEVRAIT être fait dans cette circonstance, l'agent utilisateur examinera et choisira la partie de la réponse à retenir.

Le nœud polyvalent par défaut dans un réseau DWDS intègre les trois fonctions RuleMaker RuleReserve et RuleTaker. Cependant, toute combinaison partielle de ces fonctions peut également être utile.

5.2 Mises en œuvre de référence versus des solutions logicielles particulières

DWDS n'est pas une implémentation particulière de logiciel ou de service. Il s'agit plutôt d'une spécification générale qui communique ce que toute implémentation d'une application RuleMaker, d'un composant RuleTaker et d'un service réseau RuleReserve est censée accomplir, sans restreindre la modalité.

La conception, la construction et le contrôle de systèmes fonctionnels, lorsqu'ils sont entrepris de manière itérative dans le cadre du processus de recherche de conception, aident les contributeurs à réfléchir à la composabilité de bout en bout. Une mise en œuvre de référence est un déploiement minimal d'une conception qui démontre l'opérabilité du concept, et fournit soit un échafaudage préconstruit utilisable, soit une inspiration indirecte pour une mise en œuvre de production complète. Son comportement opérationnel doit être cohérent avec la documentation actuelle du système, et celle-ci doit être entretenue au fur et à mesure de l'évolution de la conception. Les mises en œuvres de référence utilisent généralement des langages de programmation et des plates-formes optimisés pour faciliter la compréhension, la réalisation et la tenue à jour. Les mises en œuvre du véritable monde visant un usage réel seraient plutôt optimisées pour la précision, la scalabilité, la vitesse et la résilience.

5.3 Méthodes pour le computation distribué et décentralisé à haute performance

5.3.1 L'informatique rapide et lente : Externaliser le travail de computation en dehors du run-time

Le célèbre ouvrage de Daniel Kahneman, *Thinking Fast and Slow*, traite de la distinction, en psychologie cognitive humaine, entre la détermination immédiate et la délibération réfléchie. Les deux modes ont été nommés de diverses manières par différents auteurs : le système 1 / processus impulsif / automatique / heuristique versus le système 2 / processus réfléchi / réflexif / analytique (Kahneman, 2011) (Strack & Deutsch, 2004) (Bargh & Chartrand, 1999) (Evans, 1984).

Nous nous intéressons ici au trituration rapide des données déterministes avec des informations nécessaires et suffisantes, plutôt qu'à la sélection heuristique rapide en situation d'incertitude. Il y a quinze ans, dans l'introduction d'un discours-programme intitulé *The Computing Machines in the Future*, Richard Feynman a caractérisé la différence entre ces deux styles de traitement des données : « On appelle souvent cela l'intelligence artificielle, mais je n'aime pas ce nom. Peut-être que les machines inintelligentes peuvent faire encore mieux que les machines intelligentes. » (Feynman, 2005, p. 28) Nous proposons de les distinguer en tant qu'intelligence artificielle (IA) et naïveté artificielle (∅A) :

- *L'intelligence artificielle (IA)* effectue l'acquisition active de connaissances et le raisonnement inductif avec l'inférence variationnelle stochastique et l'apprentissage en double boucle (apprendre à apprendre) pour résoudre les lacunes d'information afin de guider l'action dans un degré spécifié de tolérance au risque.
- *La naïveté artificielle (∅A)* effectue une mise en correspondance passive des signaux demande-réponse avec des portes logiques et de consultation pour trier les données, et peut utiliser des opérations arithmétiques et booléennes de base pour transformer les données. Ce système « sans apprentissage » ne conserve aucune donnée de l'utilisateur.

L'extrême rapidité, le volume et la précision du traitement des données sous contrainte ∅A peuvent donner aux utilisateurs l'impression d'utiliser un système d'IA très « intelligent ». Mais en fait, il s'agit simplement d'un fonctionnement à haute performance de l'informatique « muette », analogue à : la cheville carrée dans le trou carré, la cheville ronde dans le trou rond. Il n'y a pas de prétention à l'« innovation », puisque chacune des techniques décrites est connue depuis longtemps dans l'histoire de l'informatique et n'implique guère d'autre effort que de surmonter certaines idées préconçues. En particulier, nous prévoyons que le principal obstacle à l'adoption de ces techniques ne sera pas dû à la sophistication technologique ; le défi réside plutôt dans le fait que les développeurs de systèmes informatiques créatifs et brillants de l'industrie, du gouvernement et des universités n'associent généralement pas les techniques démodées, simples, pour la plupart inutilisées et oubliées, à l'obtention de hautes performances dans les applications de la génération actuelle. Il est plus naturel, et cela se comprend, de s'enthousiasmer pour les méthodes novatrices, avancées et à gros budget (Simons, 2012).

5.3.2 Transformation du langage naturel complexe en langage naturel structuré simple

Un auteur de règles utilisant une implémentation de la spécification RuleMaker n'a pas besoin d'écrire de code de programmation. L'interface de RuleMaker aide une personne à exprimer des propositions normatives simples dans un langage naturel uniformément structuré, même si elles proviennent d'un langage naturel non structuré, et lui fournit ensuite un moyen d'établir des relations logiques dans un tableau adjacent. Pour transposer un langage naturel libre non structuré en langage uniformément structuré, il faut être capable de composer des phrases claires. Cela peut sembler manifeste, ou même condescendant, mais un tel raffinement est difficile à obtenir lorsque de multiples facteurs entrent en jeu.

DWDS RuleData spécifie une grammaire de type « phrase structurée » avec un seul formulation déclaratif. Malgré cette contrainte syntaxique très rigide, il n'y a pas de limites à la portée sémantique. C'est l'inverse de la technique plus courante du Web sémantique, qui consiste à prendre en charge des expressions complexes à l'aide de schémas sémantiques rigides (par exemple, RuleML) et de structures syntaxiques tolérantes (par exemple, SGML). Les deux approches ne s'excluent pas mutuellement ; elles sont complémentaires et peuvent être déployées simultanément.

La spécification de données avec une directionnalité laisse la gestion de la sémantique entre les mains de personnes qui ont la prérogative, la motivation, la connaissance du domaine et la familiarité socioculturelle pour adapter l'expression de chaque phrase de chaque règle, et qui sont motivées pour faire un effort réel pour fournir une reproduction fidèle de l'intention normative complète de la règle originale, et avec une distorsion minimale.

5.3.3 Externaliser la complexité linguistique de la structure de la règle, pour simplifier la fonction

La spécification « données avec une directionnalité » implique la décomposition explicite de la structure syntaxique et de la structure logique de l'expression sémantique. Le travail de computation peut alors se concentrer sur une correspondance simple et efficace des symboles. La complexité sémantique est externalisée à partir de l'exécution en demandant aux créateurs de règles de placer les mots du langage naturel dans un ensemble prédéterminé de champs syntaxiques ayant des fonctions grammaticales fixes. Collectivement, ces éléments permettent la construction d'un seul type de phrase déclarative, mais n'importe quel nombre et n'importe quelle variation de ces phrases peuvent ensuite être positionnés comme étiquettes de ligne d'une table qui fournit la structure logique et le mode normatif d'une règle. Cette méthode est expliquée en détail ci-dessous.

- *Rendre les relations logiques explicites :* Un tableau logique DWDS utilise une phrase par ligne d'un table verticale d'E/S tetranaire, reliant chaque condition d'entrée et chaque assertion de sortie aux numéros {00,01,10,11 | 00,01,10,1} ou symboles {X,√, &, ? | ⃠, !!, ○, ! }.
- *Rendre les éléments syntaxiques explicites :* La porte logique DWDS possède une « grammaire de structure de caractéristiques typées » (Wintner & Sarkar, 2002) avec six éléments syntaxiques permettant aux utilisateurs de créer des phrases déclaratives.
- *Rendre les règles facilement lisibles et efficacement computables :* Les « règles sous forme de données » de DWDS sont rédigées dans le langage naturel choisi par l'utilisateur, et le paquet de données syntaxiquement pré-paramétré est sous forme exploitable par la machine.

5.3.4 Externaliser la computabilité en requérant que l'expression des règles ne soit PAS Turing-complet

DWDS RuleData est conçu pour la programmation par tuple (Underwood, 2011), avec l'exigence obligatoire qu'il reste inférieur à ce qui est requis pour une expression Turing-complet. Il y a plusieurs raisons de s'assurer que RuleData soit incapable d'une expressivité Turing-complète :

- Il n'est pas nécessaire d'être Turing-complet pour résoudre la catégorie de problèmes abordés par DWDS. Par conséquent, les risques liés aux expressions qui pourraient être complètes en Turing sont intrinsèquement anticipés et limités au niveau de la conception.
- DWDS RuleData supporte le computation strictement déclarative. Une règle peut prendre du temps à computer, mais l'inspection peut valider qu'elle s'arrêtera. Le « problème de l'arrêt » est lié à la programmation procédurale Turing-complet.
- La nature ouverte et distribuée d'un système DWDS exige une garantie de conception intrinsèque que les attaques par injection de messages DWDS RuleData ne peuvent pas introduire de logique procédurale dans les environnements des utilisateurs.

5.3.5 Externaliser les données de contrôle et les données de relations logiques en séparant les données de la procédure

Le système DWDS sépare les données de type « quoi faire » des procédures de type « comment faire » :

Les données de relations logiques et les données de contrôle : GIVEN « a » is ; WHEN « b » is ; THEN « c » is ;

Opérations de logique procédurale impérative. IF « d » est ; THEN faites « e » ; ELSE faites « f ».

Chacun de ces éléments est une structure élémentaire de la triangulation logique.

- *Triangulation déclarative des données* : Deux entrées - une sortie :

  La déclaration logique GIVEN-WHEN-THEN relie un contexte empirique [GIVEN 'a' is] ET une circonstance empirique [WHEN 'b' is], avec une proposition déclarative conséquente [THEN 'c' is]. (North, 2006) (Fowler, 2013).

- *Triangulation de la procédure impérative* : Une entrée et deux sorties :

  Une déclaration logique IF-THEN-ELSE relie un état futur contingent [IF 'a' is], avec soit une action impérative conséquente [THEN do 'b'], soit [IF 'a' not], une action impérative par défaut [ELSE do 'c']. (Mladenic et al., 2003, p. 8-9)

5.3.6 Externaliser la lourdeur du processage des données dans des tableaux

Il est facile de perdre de vue les gains d'efficacité que le computation tabulaire peut atteindre, par rapport au computation pas à pas à partir de données d'entrée avec un code procédural. Les systèmes numériques optimisés pour le calcul en mémoire avec une porte logique et des données tabulaires sont rapides et efficaces. Les tableaux peuvent également rendre la structure logique d'un programme plus compréhensible. Art Lew a observé qu'un programme « peut être conçu comme une table de décision, être exécuté tel quel et être auto-documenté ! » (Lew, 1983, p. 183)

La transformation axée sur les données des 70 dernières années inclut l'informatique déclarative tabulaire ; par exemple, depuis le début, il est courant d'utiliser des tableaux pour les données de référence telles que les identifiants, les catégories, les indices et l'adressage est une pratique courante en informatique. Mais à notre avis, le potentiel d'un réseau mondial décentralisé de portes logiques tabulaires cohérentes sur le plan informatique n'a pas encore été réalisé. Le DWDW est notre contribution à cette quête.

La porte logique DWDS utilise trois topologies élémentaires pour les données tabulaires : produit cartésien, pile verticale et bande horizontale. Chacune partage le même schéma : DONNÉ un ensemble de faits contextuels, QUAND un ensemble de faits particuliers sont également documentés, ALORS certaines normes sont considérées comme étant « en vigueur » et « applicables ».

5.3.7 Externaliser des algorithmes réutilisables (récupération) de tables de produits cartésiens

Les psychologues cognitivistes Pierre Barrouillet et Michel Fayol ont documenté un ensemble de facteurs qui conduisent à une transformation des processus de pensée humains, du calcul procédural vers « une récupération en mémoire plus rapide et moins coûteuse des éléments d'information » dans le cerveau humain. Il s'agit de « passer d'une stratégie algorithmique ... à une stratégie de récupération directe des résultats en mémoire ». (Barrouillet & Fayol, 1998, 364-66)

La récupération rapide en mémoire peut être utilisée à grande échelle avec des ordinateurs utilisant une méthode qui est si évidente qu'elle est habituellement négligée. Il s'agit de tableaux contenant des réponses précalculées. L'écriture des formules ne doit pas être techniquement difficile, mais elle serait inutilement compliquée par rapport à une consultation instantanée d'un tel tableau s'il était disponible gratuitement sur Internet dans un format générique, et répliqué localement pour réduire la latence. Une telle consultation de la table serait presque aussi rapide qu'un *ping*. Cette méthode de faible technicité peut accélérer le traitement des données de la même manière que la plupart des gens se souviennent par cœur que « trois cinq font quinze » sans effectuer réellement l'opération arithmétique dans leur tête (Barrouillet & Fayol, 1998). Le traitement des données nécessaires à l'élaboration de la réponse est ainsi externalisé une fois pour toutes, littéralement. Lorsqu'une telle table est disponible en ligne dans un format générique, n'importe quelle application dans n'importe quel langage de programmation, exploitée par n'importe qui dans n'importe quel but, peut facilement effectuer une consultation pour obtenir la réponse, en gardant ses ressources informatiques concentrées sur la tâche principale qu'elle peut avoir.

5.3.8 Externaliser les conditions et les assertions déclaratives des relations logiques

Afin d'obtenir une structuration cohérente des règles qui serait simple à comprendre pour les humains et efficace à traiter pour les ordinateurs, le DWDS sépare et externalise les ensembles de phrases déclaratives qui expriment les conditions d'entrée et les assertions de sortie des règles, de leurs relations logiques et de leurs modes normatifs.

Une table logique DWDS utilise des conditions et des assertions déclaratives simples en langage naturel structuré comme étiquettes pour des rangs simples de données qui spécifient les relations logiques et les modes normatifs. Ceux-ci sont configurés pour fonctionner comme des portes logiques virtuelles, c'est-à-dire que les combinaisons de signaux d'entrée sont mappées de manière décisive aux combinaisons de signaux de sortie. Les auteurs de règles attribuent des symboles à chacun des deux ou plusieurs énoncés déclaratifs d'entrée, puis ils attribuent les symboles {00,01,10,11} pour associer chaque combinaison de conditions d'entrée déclaratives à une ou plusieurs assertions de sortie déclaratives, où les symboles ont la signification sémantique suivante :

Numéros binaires	Conditions d'entrée	Assertions de sortie
00	NON	PAS
01	OUI	DOIT
10	OUI ET NON (LES DEUX)	PEUT
11	OUI OU NON (PAS CERTAIN)	DEVRAIT

Cette méthode permet de contourner une grande partie de la complexité de l'expression des règles en l'externalisant vers des agents humains ou des machines. Des règles complexes peuvent être construites à l'aide d'instructions structurées simples et de signaux numériques. Le tamisage et le traitement des règles est alors simple et déterministe.

5.4 Les règles commes données avec une directionnalité

5.4.1 Structure de données des enregistrements [rule.dwd]

Dans le DWDS, un agent créateur de règles communique des assertions impératives avec des propositions normatives pour aider les agents assujettis à des règles avec des déductions empiriques. L'obligation, la permission ou l'encouragement entre les « agents » humains et les machines peuvent être communiqués avec une efficacité optimale dans un cadre dynamique complexe multi-objectifs et multi-contraintes. La transmission de l'information de bout en bout est suffisamment intuitive pour qu'une large population d'agents humains créateurs de règles et d'agents créateurs de règles puissent communiquer entre eux des propositions normatives sans avoir à connaître les méthodes formelles de traitement des données ou de programmation informatique, mais elle est également structurée de manière suffisamment précise pour être facilement analysée et traitée sur n'importe quelle plate-forme informatique que les agents créateurs de règles peuvent préférer à utiliser ou à déléguer.

5.4.2 Protocoles de transmission des données avec une directionnalité

La configuration par défaut de la connexion réseau des composants RuleMaker, RuleTaker et RuleReserve est « hypertext transfer protocol - secure » (https :) sur le port 443 du protocole de contrôle de transmission (TCP) pour la transmission réseau cryptée des messages transitoires [is.dwd] et [ought1.dwd], et le « InterPlanetary File System » (ipfs :) sur le port 4001 pour le stockage et la récupération sur le réseau des enregistrements persistants [rule.dwd] et [lookup.dwd]. Dans ce scénario, tous les messages et les enregistrements transmis se mêlent au trafic Internet général. L'Internet des règles peut être mis en œuvre avec les paramètres existants de gestion des pare-feu et du trafic Internet, et les administrateurs de réseau n'ont pas à s'occuper de configurations non conventionnelles.

Le DWDS n'exige pas, mais décrit également, pour examen, le potentiel d'un nouveau protocole Internet de transfert de données avec une directionnalité (DWDTP), désigné ici par la chaîne (dwdtp :), que nous suggérons de transmettre sur le port 7077, qui n'a pas encore été attribué. Le chemin dwdtp : est envisagé comme fonctionnant nativement avec le protocole de messagerie QUIC (Quick UDP Internet Connections) (Roskind, 2013) (Iyengar & Thomson, 2019) et employant la « négociation de protocole de la couche application » (ALPN). (Thomson, 2021)

5.4.3 Identifiants pour les ressources [rule.dwd] et [lookup.dwd]

Le DWDS exige que chaque enregistrement [rule.dwd] et [lookup.dwd] ait un identifiant unique afin de pouvoir s'y référer de manière précise, pratique et flexible. Notre conception assemble une méthode hybride concise à partir de quatre identifiants existants :

- *URI* : Context identity (Un élément est reconnaissable par sa situation) ;
- *CID* : Expression identity (Un élément est reconnaissable par ses détails) ;
- *UUID :* Inception identity (Un élément est reconnaissable par son instanciation.) ; et
- *SemVer* : Provenance identity (Un élément est reconnaissable par son ascendance).

Les URI composites qui en résultent peuvent sembler longs, mais ils sont optimisés pour une efficacité informatique tout en maintenant une validation adéquate par inspection et une lisibilité des parties autorité et version.

5.4.4 La « règle 256 » du diagnostic

La règle 256 est une règle de diagnostic générée par le système qui est conçue pour être « en vigueur » à tout moment (date/heure) et partout (globalement), pour tout (bien, service, actif) et toute action (industrie). Elle porte ce nom parce qu'elle contient les 256 permutations de l'ensemble complet des éléments tetranaires {00,01,10,11}, sans nulles, dans un ordre identique pour les conditions d'entrée et les assertions de sortie. Comme la configuration numérique des scénarios de conditions d'entrée correspond à la configuration numérique des scénarios d'assertions de sortie, l'exécution d'un [sieve.dwd] sur cette règle devrait générer une sortie identique à l'entrée.

5.5 Tamisage des données

5.5.1 Critères de particularité vs critères de conjonction

Pour l'essentiel, le tamisage des données par DWDS est une reformulation des concepts et des méthodes de classement d'il y a un demi-siècle, et ceux-ci ne sont pas uniquement basés sur l'informatique. Dans le domaine de la psychologie cognitive, on distingue deux méthodes de recherche visuelle ; le DWDS utilise d'abord l'une, puis l'autre. Dans les années 1970, Anne Treisman et Garry Gelade ont distingué la « quête de particularités », où l'attention se porte sur la présence d'une certaine spécificité de la cible, de la « quête de conjonctions », où l'on recherche une juxtaposition visuelle de plusieurs éléments. Ces auteurs observent : « il semble que nous puissions détecter et identifier des particularités séparables en parallèle sur un écran.... Les conjonctions, en revanche, exigent que l'attention focale soit dirigée en série vers chaque emplacement pertinent. » (Treisman & Gelade, 1980, p 132) Leur observation est importante pour la conception du DWDS. L'inspection parallèle rapide de plusieurs particularités individuelles de métadonnées est employée dans le RuleReserve, tandis qu'une évaluation sérielle conjointe des éléments descriptifs de données est effectuée par le RuleTaker.

5.5.2 RuleReserve utilise [is.dwd] comme [sieve1.dwd] pour trouver les [rule.dwd]s

Le réseau RuleReserve réalise trois fonctions : le stockage, le tamisage et la messagerie :

- *stockage distribué* des enregistrements [rule.dwd] sur une table [rulereserve.dwd] *n* x *m*, un enregistrement par ligne, maintenue en ligne via l'IPFS décentralisé (Benet, 2014) ;
- *un tamisage* efficace pour réduire [rulereserve.dwd] à un ensemble de rangs [rule.dwd] qui sont considérés par leurs auteurs comme étant « en vigueur » et « applicables » ;
- *une messagerie* à la demande à grande vitesse qui reçoit les requêtes [is.dwd] et envoie les réponses [ought1.dwd] ;

Lorsque le réseau RuleReserve reçoit un message de requête [is.dwd], il est censé renvoyer rapidement un message de réponse [ought1.dwd] contenant chaque [rule.dwd] de sa collection qui est à la fois « en vigueur » pour le contexte et « applicable » aux catégories décrites dans le message [is.dwd] d'origine.

Toutes les données [rule.dwd] et [lookup.dwd] sont stockées et adressées sur les nœuds RuleReserve participants dans une matrice [m x n] distribuée et délibérément redondante (c'est-à-dire *m* lignes x *n* colonnes), appelée [rulereserve.dwd]. Les données de chaque ligne indexée sont disposées comme une longue bande télex sur laquelle chaque enregistrement [rule.dwd] et [lookup.dwd] est étalé horizontalement. Le processus de tamisage peut sembler être une tâche énorme, mais il est effectué de manière massivement parallèle à travers le grand tableau décentralisé [rulereserve.dwd] qui est distribué sur le réseau IPFS. Le message [is.dwd] est préconfiguré pour fonctionner comme un [sieve1.dwd] sur la collection [rulereserve.dwd]. L'opération de tamisage proprement dite est une « recherche de caractéristiques » très simple et rapide de valeurs clés en mémoire sur un petit nombre de valeurs connues.

Une exigence obligatoire du DWDS est que le réseau RuleReserve ne soit pas capable de gérer la logique des règles. Cette prémisse ∅A minimise les capacités des opérateurs de réseau. Le traitement logique pour déterminer quelles règles sont invoquées pour la circonstance d'un utilisateur final est reconnu comme étant une prérogative préalable et exclusive des agents de prise de règles.

$$\text{SOIT + REGLE} \Longrightarrow \text{DOIVE}$$

5.5.*3 RuleTaker utilise [is.dwd] et [rule.dwd], créant [sieve2.dwd] pour obtenir les assertions 'invoquées'*

Dès qu'un composant RuleTaker reçoit un message de réponse [ought1.dwd], un processus de données en plusieurs étapes est lancé pour réduire les données des règles « en vigueur » et « applicables » pour obtenir les relations SOIT + REGLE, aux règles « invoquées », qui résolvent la relation SOIT + REGLE $\Longrightarrow$ DOIVE.

Il peut y avoir plusieurs façons de mettre en pratique la séquence DWDS RuleTaker. La spécification est intentionnellement conçue pour être construite sur diverses plateformes, et pour rester flexible aux préférences des programmeurs, et aux méthodes disponibles dans différents langages de programmation et de notation mathématique.

Bien que les humains s'attendent à des phrases complètes en langage naturel, l'unité de données essentielle avec laquelle un programme informatique déclaratif travaille est un triple sujet-prédicat-objet (SPOT). En d'autres termes, lorsque « Le conteneur d'expédition est rempli », nous avons des données impliquant un sujet (« le conteneur d'expédition »), un prédicat (« est ») et un objet (« rempli »). C'est ce qui rend le table verticale d'E/S compréhensible, même sans phrases complètes.

Les artefacts [rule.dwd] et [is.dwd] consistent en des faits déclaratifs fonctionnellement inertes, sans code procédural. De plus, le DWDS exige que RuleTaker soit incapable d'exécuter toute procédure exprimée directement dans une assertion de sortie ou appelée à partir d'une ressource [lookup.dwd]. D'un autre côté, rien n'interdit aux utilisateurs finaux d'utiliser de telles fonctions procédurales, à condition qu'elles soient exprimées et mises en œuvre via une plate-forme externe, adjacente ou sous-jacente, et jamais dans un composant ou une application DWDS.

La conception du DWDS permet au RuleTaker de puiser dans une bibliothèque interne limitée de méthodes statiques simples pour les transformations de données, les opérations booléennes et les fonctions arithmétiques, toutes à signature numérique, qui peuvent être employées dans des phrases exprimant les conditions d'entrée et les assertions de sortie d'un enregistrement [rule.dwd].

Chapitre 6 : Un mise en œuvre de référence pour la validation du concept

Cette section fournit des détails sur le développement des composants de logiciel opérationnel basés sur la conception du DWDS. La présente entreprise de recherche sur la conception a impliqué diverses mises en œuvre partielles de la spécification du système DWDS en quatre parties, à savoir le portée et structure de données du RuleData, la conception de l'application RuleMaker, la conception du entreposage et de la mise en réseau des données RuleReserve et la conception du composant RuleTaker. La première spécification complète de bout en bout a été mise en œuvre au cours du premier semestre 2022, sur la base des descriptions de la spécification DWD de décembre 2021 (c'est-à-dire une ébauche antérieure de cette thèse). Le premier logiciel de mise en œuvre de référence fonctionnel basé sur la version 3.x de la conception du DWDS a été démontré lors de sessions publiques en ligne en mars 2022.

Ces implémentations fournissent :

(a) la preuve que la conception décrite dans la thèse de décembre 2022 est réalisable ; et,

(b) la preuve que le DWDS est jugé « utile » (digne d'être mis en œuvre) par certains praticiens.

La programmation interprétative d'un logiciel de « classe production » à partir d'une nouvelle spécification est un processus méthodique et itératif, qui exige de la patience pour les détails et un engagement considérable. Cette mise en œuvre de référence complète a été menée par deux professionnels externes sur la base du volontariat, collaborant publiquement sous des licences libres/libres (Apache 2.0 ; Affero GPL). Chaque composante essentielle est décrite dans la thèse en quatre sections : Vue d'ensemble ; Mise en œuvre antérieure ; Détails de la mise en œuvre actuelle ; et, Expérimentation.

Chapitre 7 : Conclusion

7.1 L'objectif et le résultat obtenu

L'objectif de ce travail était de concevoir une méthode nouvelle et pratique permettant à toute personne ou organisation d'écrire, de rédiger, publier, découvrir, obtenir, scruter, prioriser et, avec l'accord des parties prenantes directes, automatiser des règles sur tout réseau numérique, avec précision, simplicité, échelle, rapidité et résilience, tout en respectant les prérogatives, les accords et les préférences. Cette recherche conceptuelle a permis de produire la conception de bout en bout et les spécifications des composants d'un « Internet des règles ».

Cette recherche a été entreprise pour résoudre un problème général de communication normative à l'aide d'une solution informatique polyvalente, mais ce travail est institutionnellement situé dans le programme académique de gestion de projet d'une école d'administration. Dans ce contexte, cette recherche comble une lacune dans la théorie et la pratique de la gestion de projet qui concerne la façon dont toute partie prenante découvre et obtient, dans son langage naturel préféré, des informations factuelles sur les règles significatives qui sont « en vigueur » pour les dates/heures et les prérogatives relatives aux identités et aux juridictions d'un contexte donné ; qui sont « applicables » à la catégorie d'effort et de tâche entreprise ; et qui sont « invoquées » par une circonstance particulière du moment. Le résultat de la résolution de ce problème dans le contexte de la gestion de projet est une spécification et une mise en œuvre de référence qui résout l'objectif général : *« L'agent A, interagissant avec l'agent B, a besoin de connaître une ou plusieurs règles gérées de manière externe par les agents C..n qui sont « en vigueur » pour des contextes donnés, et sont « applicables » à un ensemble de catégories d'événements, et sont « invoquées » par des circonstances particulières » dans l'incertitude sur les agents et sur les règles.*

Cette thèse explore les fondements conceptuels des règles et des systèmes de règles, et décrit le raisonnement, la conception, la faisabilité, la généralisation et l'utilité d'une méthode informatique en réseau permettant à quiconque d'écrire, rédiger, publier, découvrir, obtenir, scruter, prioriser et, avec l'accord des parties prenantes directes, automatiser des données normatives qui relient ce qui « soit » à ce qui « doive » être, en respectant les prérogatives, les accords et les préférences. Nous proposons le Data With Direction Specification (DWDS) comme une spécification pour une classe de pipeline de processus de données résumée par la relation SOIT + REGLE $\Longrightarrow$ DOIVE.

La recherche appliquée a été poursuivie en parallèle pour valider le concept par le développement d'un logiciel de mise en œuvre de référence fonctionnel sous licence et méthodes libres. Cela a inclus une révision active par les pairs de la conception itérative avec des professionnels de diverses communautés de mise en œuvre potentielles. Les mises en œuvres de référence sont des modèles de travail minimaux des spécifications, y compris une application RuleMaker, un composant RuleTaker et un service réseau RuleReserve. L'exécution de ces éléments ensemble permet l'émergence d'un « Internet des règles ». Les exemples de cas d'utilisation qui ne sont pas ceux qui illustrent divers détails fonctionnels dépassent le cadre de la présente recherche de conception, qui vise à mettre en place un système à usage général.

7.2 Diverse contributions originales et restaurations utiles

Le thèse décrit les nouveaux éléments conceptuels et fonctionnels d'un Internet des règles. Cette section énumère deux douzaines de contributions originales [O] en matière de conception et de méthodologie, et une demi-douzaine d'autres restaurations « utiles » [U] de concepts et de méthodes de conception existants qui ont été négligés ou oubliés dans la plupart des domaines de l'informatique. Les éléments sont référencés dans l'ordre où ils apparaissent dans la thèse.

7.5 Limitations de cette recherche

Ce ligne d'investigation théorique se limite aux fondements conceptuels et à la recherche formelle sur la conception, avec seulement une portée très superficielle dans le travail de terrain simultané et autonome qui a été en cours dans divers domaines. Au cours de cette recherche, dont la licence est 100% libre/libre/ouverte depuis le début, des équipes collaboratives ont vu le jour et sont devenues très actives dans la mise en œuvre du DWDS. Un échantillon de 'use cases' externes à cette recherche conceptuelle est fourni à l'annexe C. Le DWDS et ses éléments fonctionnels sont auxiliaires, ce qui rend difficile de partager ici des informations sur les exemples de projets réels sans avoir à se plonger trop profondément dans les détails des projets eux-mêmes, et dans ces détails, perdre l'intrigue qui est au centre de la présente recherche en conception. La portée et la profondeur de cette thèse sont déjà à la limite de la complexité, c'est pourquoi il a été décidé de fixer les limites pour n'inclure que la théorie et la conception. Le thèse laisse les mises en œuvre dans le monde réel qui sont en train de germer dans l'industrie et le gouvernement à la catégorie « recherche future ».

7.6 Recherche future

7.6.1 Diverses suggestions reçues

Ce travail ouvre de nouvelles voies de recherche analytique. Lors de l'atelier inaugural « Rules-as-Data » d'avril 2022, auquel ont participé des universitaires de dix universités, certains sujets d'exploration académique future ont été soulevés lors de la session consacrée à ma présentation et à mon article sur le DWDS. Il s'agissait d'une typologie des modèles logiques de règles, des types d'incertitude dans l'expression des règles, des critères de pondération du double objectif d'accessibilité générale des règles et de légalité formelle, et de l'importance comportementale du rapport signal/bruit dans la communication entre les créateurs de règles et les créateurs de règles.

7.6.2 Recherche en cours sur les méthodes techniques

Parallèlement au domaine de la gestion de projet, il y a des questions de recherche technique intéressantes et utiles à résoudre au fur et à mesure de l'avancement des travaux sur les mises en œuvres de référence des applications RuleMaker et RuleTaker et du réseau RuleReserve. Les mises en œuvres de référence des logiciels et du réseau fournissent une base pour un large éventail de recherches intéressantes et utiles sur la conception.

7.6.3 Linguistique informatique améliorée pour la transformation numérique

La réconciliation des portes logiques DWDS avec les conseils RuleSpeak existantes devrait se faire bientôt. Ceci est pertinent pour les discussions avec les responsables du secteur public concernant l'aspect pratique de la conversion d'un grand volume de règles conventionnelles en langage naturel, vers phrases structurées en langage naturel avec des portes logiques conformes au DWDS. Il s'agit également de savoir comment mettre en place des boucles de rétroaction expérimentales afin que la capacité de transcription d'un Internet des règles prolifère par auto-organisation.

7.6.4 Faire preuve de sens commun pendant une période de « paradigmes incommensurables » .

La « spécification des données orientées » a été finalisée au moment où les années 2020 ont fracturé chaque communauté (sur la Terre, semble-t-il) en deux (ou plus) paradigmes incommensurables.

Dans *La structure des révolutions scientifiques*, le philosophe défenseur d'une « science normale » Thomas Kuhn a décrit un paradigme comme un cadre cognitif partagé d'interprétation, d'explication, de validation et d'attente qui offre à une société un degré de consensus sous-jacent sur les structures, les processus, le savoir-faire et les règles qui guident le comportement et façonnent l'avenir. (Kuhn, 1962)

Le philosophe Paul Feyerabend, défenseur d'une « science pluraliste compétitive », affirme que *la Structure des révolutions scientifiques* de Kuhn décrit un processus sans direction : « Il n'a pas réussi à discuter du but de la science » (Feyerabend, 1970, 201). Feyerabend illustre sans détour le problème : Dans le modèle de paradigmes de Kuhn, les règles sont des instructions méthodologiques pour résoudre des casse-tête. La vision de la science de Kuhn ne prévoit aucun rôle pour la direction normative en général, ni pour la vertu déontique en particulier. Bien qu'elle puisse être rigoureuse, elle est sans but.

Aujourd'hui, dans les années 2020, les personnes qui se trouvent dans des paradigmes incommensurables par rapport à leurs collègues et à leurs communautés sont néanmoins confrontées à la négociation de décisions continues au niveau micro pour la gestion quotidienne de projets, de programmes, de portefeuilles et de plates-formes, ainsi qu'à la gestion des infrastructures, des chaînes d'approvisionnement, des services essentiels et des écosystèmes de base au niveau macro dont chacun dépend.

En 1970, les adversaires philosophiques Kuhn et Feyeraband se sont mis d'accord sur le fait qu'un nouveau langage - sans impliquer un langage « neutre » - serait une étape préliminaire essentielle. C'est ici que le pragmatisme souligné dans la section méthodologique 2.1 nous a permis d'utiliser le point théorique précédent comme pivot pour résoudre en pratique une catégorie générale de problèmes du monde réel. La spécification « données avec une directionnalité » fournit une infrastructure générale de niveau méso (un « Internet des règles ») pour soutenir la communication sur la voie à suivre pour traverser cette période de désordre et de confusion.

Le DWDS RuleData, opérationnalisé dans l'application RuleMaker, est conçu comme un aide à usage général avec une structure syntaxique de contraintes très basiques, flexible à configurer, avec une gamme sémantique illimitée, et une tolérance de la langue vernaculaire, pour permettre la transmission d'informations liées à n'importe quel domaine, qui peuvent être exprimées simultanément dans n'importe quel langage, pour refléter n'importe quel paradigme.

Au milieu du choc des paradigmes qui a fait surface au début des années 2020, il y a un besoin vital de recherche participative pour l'élaboration d'un sens commun. Les méta-règles décrites dans la section 3.3.2 peuvent être propres à chaque paradigme tel que l'entend Kuhn. Mais tout comme des personnes d'idéologies complètement différentes regardent dans la même direction un lever de soleil, et préfèrent s'abriter de la pluie, la section 3.1 de cette thèse présente une règle comme toute relation directionnelle pratique, logique, éthique et esthétique communiquée entre deux ou plusieurs personnes pour associer ce qui 'soit' et ce qui 'doivet' être : SOIT + REGLE $\Longrightarrow$ DOIVE. Cette approche pluraliste de l'expression normative, tolérant l'interprétation, l'opposition et un certain degré de non-conformité, peut permettre la négociation d'un sens commun entre des paradigmes incommensurables.

Contents

Acronyms, List of Tables and List of Figures

Acronyms

ABM	Agent-Based Modelling
ALGOL	ALGOrithmic Language
ASCII	American Standard Code for Information Interchange
API	Application Programming Interface
ALPN	Application-Layer Protocol Negotiation
ARG	Argument
AI	Artificial intelligence
A∅	Artificial naïvety
ACM	Association of Computing Machinery
ADS	Automated Driving System
BNF	Backus Normal Form
BV	Bound Variable
BPMN	Business Process Modelling Notation
CQL	Cassandra Query Language
CSV	Comma-Separated Values
CODASYL	Committee on Data Systems Languages
COBOL	COmmon Business-Oriented Language
CBOR	Concise Binary Object Representation
CID	Content Identifier
UTC	Coordinated Universal Time
CORFO	Corporación de Fomento de la Producción
DWDS	Data With Direction Specification
DWTTP	Data With Direction Transfer Protocol
DMN	Decision Model and Notation
DT	Determiner
DHT	Distributed Hash Table
DSL	Domain-Specific Language
DDT	Dynamic Driving Task
DYNAMO	DYNAmic MOdels
ERA	Earth Reserve Assurance
ECA	Event-Condition-Action
XML	eXtensible Markup Language
FTP	File Transfer Protocol
HTTPS	Hypertext Transfer Protocol - Secure
I/O	Input-Output
IEEE	Institute of Electrical and Electronics Engineers
ICCPM	International Centre for Complex Project Management
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
IPMA	International Project Management Association
ISIC	International Standard Industrial Classification
ITU	International Telecommunications Union
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IoR	Internet of Rules
IoT	Internet of Things
IPFS	InterPlanetary File System
JJ	Adjective / arithmetic
JSON	Java Script Object Notation
KISS	Keep it Simple Stupid
LKIF	Legal Knowledge Interchange Format
LKB	Linguistic Knowledge Builder
LISP	LISt Processor
LSLT	Logical Structure of Linguistic Theory
MABM	Macroeconomic Agent-based Model
MD	Modal Normative Verb
NETL	NETwork Language
NN	Noun
OMG	Object Management Group
OECD	Organization for Economic Cooperation and Development
IN	Preposition
PCN	Process Chain Network
PLC	Programmable Logic Controller
Prolog	PROgrammation en LOGique
QUIC	Quic UDP Internet Connections
RaD	Rules as Data
RADS	Rules-oriented Autonomous Decentralized System
RPP-PPR	Rail-Powered Property – Property-Powered Rail Open Market Development Model
RFC	Request for Comment (within the IETF)
ReqIF	Requirements Interchange Format
RASE	Requirements-Applicabilities-Selection-Exception
RDF	Resource Description Framework
RD	Rule Data Model
RIF	Rule Interchange Format
RuleML	Rule Markup Language
RFSG	RuleFiniteStateGrammar
RM	RuleMaker
RR	RuleReserve
RT	RuleTaker
SemVer	Semantic Versioning
SWRL	Semantic Web Rule Language
SBVR	Semantics of Business Vocabulary and Business Rules
SMTP	Simple Mail Transfer Protocol
SDL	Standard Deontic Logic
SGML	Standard Generalized Markup Language
SQL	Standard Query Language
SPOT	Subject-Predicate-Object Triple
TCP/IP	Transmission Control Protocol/Internet Protocol
MC12	Twelfth Ministerial Conference, WTO
SUBREI	Undersecretary of International Economic Relations, Ministry of Foreign Affairs of Chile
URI	Uniform Resource Identifier
URN	Uniform Resource Name
UN/ CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
UNSPSC	United Nations Standard Products and Services Code
UBL	Universal Business Language
UUID	Universal Unique Identifier
UI	User Interface
VBN	Verb past participle
VBZ	VerbAuxiliary
WebRTC	Web Real-Time Communication
WTO	World Trade Organization
W3C	World Wide Web Consortium

List of Tables

List of Figures

Acknowledgments

Dr. Stéphane Gagnon at Université du Québec en Outaouais has very generously and patiently guided this research undertaking towards the sort of rigour that only a doctoral research programme imposes. I am very grateful that he saw the potential in my early concepts, contributed a wide variety of prompts for me to consider along the way (most recently, a pointer to the work of Nuer Belnap's "four-valued logic"), and helped me to achieve the overall academic—pragmatic synthesis which suits a DBA, versus a PhD thesis. Both he and Dr. Véronique Nabelsi have respectfully negotiated a way for me to focus on rule systems design, which is considered to be at the meso-layer (rule systems) of a three-tier micro-meso-macro 'project ecology' framework.

Members of my academic jury provided very useful feedback on the penultimate draft. My thanks go to Dr. Hamed Motaghi and Dr. Raphaël Khoury at my own university, and Dr. Nhamoinesu Mtetwa in the financial services sector. It was very helpful that Dr. Khoury undertook a meticulous review to track down and advise me of issues that ranged from major to minor. Two other faculty members at my university also provided comments on an earlier version, which led to significant re-organization and some expansion. Responding to all of these comments did not require any changes in the Data With Direction Specification (DWDS) functional system design emerging from the core of this work, but all their feedback greatly improved the background and explanation. I am solely responsible for all remaining gaps and gaffs.

Special thanks are due to my three earlier research supervisors at other institutions. Dr. James J. Kay, my research supervisor some years ago in Systems Design Engineering at University of Waterloo, provided me a foundation in formal ecosystem science and information theory. Dr. Geoff Harcourt, my M.Phil. research supervisor at Cambridge University guided my learning in theoretically-rigorous applied economics. And Dr. Kari Polanyi Levitt, my undergraduate research supervisor at McGill University, grounded me with human-centred systems design, and input-output data processing structures. I remain indebted to all three for their insights and generosity.

This doctoral research began in 2013 and then received a boost through financial assistance as well as substantive know-how about large-scale rules automation systems from William Olders, President and CTO of DataKinetics. It was in conversation with Bill in late 2015 that one of us named the system emerging from this specification: "an Internet of Rules". I think it was me. He thinks it was his phrase. I can't say for sure! So credit is respectfully shared. His colleague Wayne Cunneyworth, author of *Table Driven Design: A development strategy for minimal maintenance information systems*, published in the 90's by DataKinetics, introduced me to and shaped my understanding of tabular declarative programming. Thanks also to Wayne for suggesting that I consider four-element logic. In 2016, Randy McCoy, Alan Zander and Larry Strickland at DataKinetics contributed many ideas to that early 'Internet of Rules' concept and it's potential across diverse market sectors and use cases. The research funding, and this team's collaboration in the creation of Xalgorithms Foundation as a special purpose not-for-profit vehicle for management of the free/libre licensed reference implementation, enabled me to involve a wider team, and thus to get technical experimentation done at a pace that could not have happened otherwise.

Don Kelly, a long-time friend and colleague collaborating on free/libre efforts, has very generously and thoughtfully contributed deep technical insight and original design elements into the data processing pipeline that comprises the operational specification and software components – all while working his day job helping

to enhance the developer environment at Shopify, and building a young familly. Calvin Hutcheon, who recently graduated from Maryland Institute College of Art (MICA), has helped with conceptualizing the end-to-end system integration, as well as creating the system user interfaces for the applications and the website to share and seek feedback on the work-in-progress. Ryan Fleck, who recently graduated from Computer Engineering at University of Ottawa, and is now with the Cognitive Process Transformation team at IBM, developed several iterations of the RuleMaker application through the first two versions while requirements from me, Don, Calvin and Bill were fluid. He enlisted some occasional assistance from co-student Max Chen. Their patience enabled testing of rule authorship during our emerging requirements trajectory. Ryan and Max's implementation was then re-written and advanced by Ted Kim, outside of his day job as a data scientist on the regulatory side of Health Canada, and Huda Hussain, who is currently in the Bachelor of IT (Interactive Multimedia & Design) program at Carleton University. With Ted's guidance, Huda then did most of the detailed work towards creating the full-featured version 3.x of RuleMaker. For some additional explorations in a use case related to large-scale data wrangling (i.e. LandSat and ESA data) I appreciate the technical assistance of Ted, and also Deja Newton, who recently graduated from Environmental Sciences at University of San Francisco. Deja also set the 'personal journey' style for our team blogging via the Xalgorithms website.

I appreciate the help of several who provided the different computational ways of expressing a sample rule that are presented in my Section 4.1, namely Don Kelly of Shopify for the PASCAL code; Theodiris Mitsikas of the National Technical University, Athens for the RuleML; William van Woensel at Dalhousie University for the Notation3; Bernd Rücker of Camunda Inc. for the DMN table; and Ron Ross for the RuleSpeak expression. Without their help I could not have been sure that each example was being expressed elegantly.

A similar note of gratitude is due to the two dozen people who provided the 25 natural language ways of expressing a sample sentence in my Table 14. Each was very patient with me as we communicated back and forth to ensure that I had the syntactic structure of each part of that sentence correctly categorized, and so that I had the terms and phrasing accurately interpreted to suit natural expressions.

Jacob Kelter, a doctoral candidate in the combined Learning Sciences and Computer Science program at Northwestern University, is to be credited with my including an agent-based model platform among the DWDS target reference implementations. He supervised undergraduate interns Jacob Wit, Will Conboy, and Amanda Sugiharto, who were each very helpful, with the guidance of Dr. Uri Wilensky, creator of NetLogo. Jacob also led Uri and I to our joint receipt of the “Best Paper Award” at the 2022 Annual Conference of the Computational Social Science Society of the Americas, now to be published by Springer in 2023. (Kelter, Wilensky & Potvin, 2022)

Craig Atkinson, currently a research fellow at Stanford Law, and previously at the World Trade Institute, University of Bern, Switzerland, has masterfully shepherded linkages with cross-border trade and algorithmic law use cases. He cultivated a significant global community of thoughtful people, which then gave rise to pivotal feedback and insights from many others throughout the past four years. Due to Craig's multiple articles in business literature in which he built upon my “Internet of Rules” design (Atkinson, 2018a) (Atkinson, 2018b) (Atkinson, 2018c), as well as various co-authored pieces (Atkinson & Schubert,

2021) (Atkinson & Potvin, 2022a), there arose collaborations with several others exploring use cases related to international trade. Some of these linkages generated through Craig's initiative follow below.

Moritz Hessler, Co-Director of the Foreign Trade Strategy, Digitalization, IT Projects team at BMW (Munich) invited a team of MBA students in the School of Management, Economics, Law, Social Sciences, International Affairs and Computer Science at University of St. Gallen, Switzerland to prepare a *"Position Paper: Is the Internet of Rules an appropriate solution to overcome BMW Group's challenges in the area of tariff classification?"*. I am grateful for Moritz's initiative and the work of this student team: Valentin Hinum, Hanna Adi Gol, Dario Pedone, Vadim Tschanz and Pascal Zuberbühler. (Hinum et al., 2022)

Dr. Fred Olayele and Dr. Yiagadeesen (Teddy) Samy at Carleton University have been exploring the utility of my Internet of Rules design to enhance operational performance of the African Continental Free Trade Area (AfCFTA), which came into effect on 1 January 2021. With the assistance of Tokini Briggs, they recently published through Routledge the book '*Sustainable Development in Post-Pandemic Africa: Effective Strategies for Resource Mobilization*', which includes a chapter by Craig and I: '*Implementing the African Continental Free Trade Area: A Simple, Scalable, and Fast Computational Approach for Algorithmic Governance*'. For us to have that chapter published just as I was finishing off this dissertation provided me an opportunity to summarize the full 'version 1.0' specification in the context of trade agreements. Previous published works had to rely on the partially completed design.

Dr. Alessia Damonte, Associate Professor (Political Science), University of Milan, and Dr. Giulia Bazzan, Postdoctoral Researcher, University of Copenhagen (Department of Food and Resource Economics), kindly invited Craig and I to present and discuss the 'Internet of Rules' concept at their inaugural *'Rules as Data Workshop"* within the European Consortium for Political Research (ECPR) Joint Sessions, convened in April 2022 (Damonte & Bazzan, 2022). Following that event, our summary article entitled *"The Data With Direction Specification (DWDS): A 2022 Research & Development Update"* was accepted for publication in 2023 by the editors of *Regulation & Governance* (Wiley, 2022). This also provided an opportunity to outline the 'version 1.0' specification in relation to legislation. Thanks to Christopher Frantz, who I met through the *Rules-as-Data Workshop* for his prompt and thoughtful suggestions on my proposed typology of 'Controlled Natural Language' frameworks. That conversation and the categorization are 'in progress'.

Nicholas Schubert, while he was in the Government of Chile's Subsecretaría de Relaciones Económicas Internacionales (though he is now at Google), very proactively pursued the trade facilitation use case based on my Internet of Rules design in the contexts of both internal government legislative procedures and international trade agreements. He and Craig published a book chapter about it. (Atkinson & Schubert, 2021) More recently Christine Weidenslaufer in the Biblioteca del Congreso Nacional (BCN), Chile has been exploring the use of my design to enhance the general accessibility and usability of legislation, ranging from regulatory through to constitutional law. In addition to her BCN role, she is presently also enrolled at the Global Professional LLM Program in Innovation, Law and Technology, University of Toronto.

Hiroki Habuka, while he was Deputy Director for Global Digital Governance, Ministry of Economy, Trade and Industry (METI), Japan (he is has returned to private practice as an attorney), integrated my emerging Internet of Rules concepts into the Japanese government's human-centred 'Society 5.0' strategy. This combines goals shared among stakeholders with methods implemented by distinct stakeholders. In their approach, government is a facilitator of multi-stakeholder rule-making, but is not the sole provider of rules; also it is understood that rules are not static—they need to be evaluated and improved. Hiroki's thoughtful probing and feedback led to some useful improvements, including an approach scalable Internet of Rules introduction described in Section 7.4.3.

Throughout early conceptualization of the DWDS design, when thinking about practical uses cases was needed in order to conceptualize a general purpose 'Internet of Rules', Ken Holman generously allocated time to answer many questions. He is a member of the W3C Working Group that developed XML from SGML, was founding chair of the two OASIS XML and XSLT Conformance Technical Committees, and at various times has been co-chair of the UBL Technical Committee (Universal Business Language). In follow-up to my participation in the inaugural W3C workshop on Web payments in 2014, I prepared with Ken's essential guidance: *"A Quick Introduction to UBL Oriented to Payment Solutions Designers"* (Potvin, 2015)

The late Harold Boley, pioneer in Semantic Web rule languages and lead designer of the RuleML schema, also provided guidance during early explorations, and he invited me to present to the RuleML monthly online workshop on 28 June 2019. In the second week of April 2022 I sent a question to a RuleML community list, which led to the opportunity of an extended discussion with the preeminent historian of decision table methods, Jan Vanthienen, professor of information systems in Belgium at KU (Katholieke Universiteit) Leuven, Department of Decision Sciences and Information Management. Prior to this conversation I had already cited six of his papers in this dissertation. During this call Dr. Vanthienen emphasized the critical distinction between a 'decision table', which must have only one outcome, and my DWDS logic gate which can potentially have more than one outcome when the Yes-AND-No and Yes-OR-No elements are in use among the Input Conditions. In practice, this means the decision is the prerogative of, or is delegated to, a human end user.

A nod of thanks is due to Hubert Laferriere, Director of the Advance Analytics Laboratory, Operations Planning and Performance Branch, Immigration, Refugees and Citizenship, Government of Canada, for inviting me to speak about work-in-progress on the Internet of Rules concept as part of the Algorithmic Decision-Making Round Table during the *Symposium on Algorithmic Government*, Ottawa, April 23-24, 2019. That panel included my research supervisor Dr. Stéphane Gagnon, University of Québec, Outaouais, and Dr. Gregory Richards, Adjunct Professor, Telfer School of Management, University of Ottawa. (Government of Canada, 2019)

While working on the final edit of this dissertation in November and December 2022 I had the good fortune to meet a specialist in logic evolution, syntax, semantics and computation. Through a few brief conversations with Dr. Magdalena Pradilla Rueda (Pradilla Rueda, 2008), and some time with her book *Lógica Basica: reflexiónes epistemológicas, históricas y filosóficas, Vol. 2, Pt 1* (Pradilla Rueda, 2017), I was guided to reflect more carefully on the different philosophical foundations of the various logic schemes in my Section 4.2 "Alternative Logic Data Models". As a result that section has been improved, and I have added an entry to Section 7.4.1 on future research.

Also in those final two months of 2022, the OpenAI/GPT-3 (Generative Pre-trained Transformer) beta service became available online. (Vaswani et al., 2017) (Brown et al., 2020) (OpenAI Inc., 2021). At that time already revising my Section 4.2 based on what I had learned from Dr. Pradilla Rueda, I also experimented with this new service on some of my logic model distinctions. I found that it was essential to assess results very carefully because, for example, when I asked for academic journal references to support what was provided in some of its replies, OpenAI/GPT-3 often put forward entirely fabricated articles citations and abstracts. It excused these as ‘simulations’ when I challenged it. (This intermingling of simulated and genuine replies reminded me of the ‘magical realism’ in Gabriel Garcia Marquez’ novels!) With that caveat in mind, some valuable refinements in my distinguishing and characterization of alternative logic frameworks did emerge from this human-machine dialogue, as attributed in my text. (Potvin & OpenAI/GPT-3, 2022). OpenAI/GPT-3 is a sharp tool, to be used with care.

I am grateful to several people who, a decade ago in 2013, saw some merit in my early design concepts, and through discussion, contributed to how those ideas took shape. Immediately after the mid-September 2013 EDGE-NYC Conference (The Economist Group, 2013) , an informal follow-up “Web Payments Lunch and Learn” session was kindly arranged for me by Manu Sporny, principal designer of JSON-LD, and Eric Anderson, Lead Data Scientist at Bloomberg LP. My doctoral program was getting underway at Université du Québec, and this provided an opportunity for me to discuss my initial research pursuit in the context of algorithmic transaction management. I’m also indebted to Manu for his subsequent collaboration on a joint submission (Sporny & Potvin, 2014) in which we invited the US Federal Reserve to participate in the inaugural W3C Meeting on Web payments (W3C, 2014). where I presented and discussed a practical use case for algorithmic transaction management (Potvin, 2014b). Then Connie Theien, Senior Vice President (Payment Industry Relations) at the Fed accommodated my participation in their two-year Faster Payments Task Force (FPTF). (Faster Payments Task Force, 2017).

Shelley Sandiford, via her company Sciconic Science Media, produced a 3-minute animated video which helped us conceptually position an Internet of Rules at the Solutions Showcase within the US Fed’s Task Force. Shelly’s video retained such relevance 5 years later that, with merely a few name changes, we re-released it in 2021. (Sandiford, 2021)

Also in the context of the Fed’s task force, I thank Stan Stalnaker, founder of Hub Culture Inc., and David Walker, (then) President of the not-for-profit Electronic Clearing House Organization (“ECCHO”), for inviting me to integrate my emerging functional idea of an Internet of Rules for improved conformance management into their joint submission describing next-generation end-to-end payments solutions. (Stalnaker et al., 2016) (Stalnaker, 2016)

Initial technical explorations with operational software began in 2016, when I had a general functional concept, but did not yet have either a system boundary or a design specification. Michael Richardson, Samir Hussain, Patrick Naubert, Simon Deziel, Francis Bordeleau and Simon Redding were quite patient in helping to undertake design experiments at that stage. Similarly the initial conceptual positioning (branding) at that time was assisted by Mario Godbout, Stephen Hards, Fred Sune, Peter Gabany, Colin Elliott, Dom Proux and editor Juan Caicedo. In regard to early positioning, Richard O’Brien helped with valuable discussions, and he introduced me to the US representative to UNCITRAL. David Hamilton presented my Internet of Rules

concept in 2016 at UNCTAD 14. (UNCTAD, 2016) Also Omaira Mindiola provided occasional administrative support (and let me portray her as a dockworker in Figure 27).

This note of acknowledgments would be incomplete if I did not express appreciation to all the good people who have created the free/libre/open licensing, methods, software and frameworks that the DWDS implementations depend upon to deploy as working applications. Richard Stallman, Founder and President of the Free Software Foundation has always replied promptly to help me refine licensing and contributor arrangements that require careful thinking, and are easy to overlook. He also graciously accepted to co-present my early Internet of Rules design to Members of the Canadian Parliament, senior government decision-makers, and union executives, in order to simplify and semi-automate collective agreements.

The free/libre/open software community shows up in unexpected places. In 2015 the Hon. David Graham became the first Member of the Canadian Parliament with first-hand technical software programming, systems administration and free/libre/open licensing knowledge. He promptly arranged for me and colleagues working on the early reference implementation components for an Internet of Rules to make a presentation on 16 May 2017 to Canada's "House of Commons Standing Committee on Operations & Expenditures". An earlier conversation with David led to the particular design decision, explained in Section 4.3, that rules-as-data (and rules-as-code) expressions should not replace or be inserted into legal documents. He explained that this belongs in a 'schedule' or other type of attachment to legal text. It should remain subordinate to the formal legal language text endorsed by legislators or parties to the agreements.

My indirect but very genuine thanks goes to all of the authors past and present who are cited in the Reference List. The effort of undertaking and completing this dissertation gave me a whole new degree of respect for what has gone into every original design contribution that I have learned about through all of these articles and books.

Family members have have assisted with good ideas, while being exceptionally patient. Angela Bernal provided ongoing thoughtful feedback on numerous elements while this improbable project evolved from notional to operational. Julian thought up the term 'Oughtomation', the first name for what was later to become the 'Data With Direction Specification' (DWDS), and he helped weed the thought garden. Sara suggested the pedagogical potential of RuleMaker, and its use for aviation rules.

All responsibility for errors and omissions that remain in the work rests with me, of course. So I invite you to let me know of flaws that you may see, and of any inspirations arising from this work that you can share.

Joseph Potvin
potj09@uqo.ca
jpotvin@xalgorithms.org
https://xalgorithms.org
https://gitlab.com/xalgorithms-alliance
https://xalgorithms.redminepro.net
@xalgorithms
@jrpotvin

Chapter 1: Introduction

1.1 Objective

This research fills a gap in project management theory and practice,[1] which concerns how a project stakeholder is presumed to discover and obtain factual knowledge of the significant rules that are 'in effect' for dates/times and prerogatives relating to identities and jurisdictions of a given context; that are 'applicable' to the class of endeavour and task being undertaken; and that are 'invoked' by a particular circumstance of the moment. Practical people, as individuals and also on behalf of entities, need to obtain timely and comprehensible awareness of relevant rules in order to exercise judgment: to actively aligning with a particular rule, or to decide not to.

The "project manager" role was defined more than a half century ago by Paul Gaddis to be a person seeking to achieve a planned outcome "on time, within budget, and in conformance with predetermined performance specifications" (Gaddis, 1959). A project is any undertaking with a declared objective pursued within a set of physical, human, financial and time constraints. These descriptions hold for relatively simple projects, as well as for complex ones involving outcomes and performance goals that are known to be tentative, responsive and approximate. They are equally relevant to series of projects that may be considered programs or services, and to concurrent sets of projects that comprise a portfolio.

A "rule" is a canon or precept by which repeated behaviour is guided through authority, agreement or preference. Rules are expressed as assertions with the modal auxiliary verbs shall (must), should (ought to) or may (can). Every human agent undertaking a project is both a rule-maker, and a rule-taker, and every human endeavour shapes and is shaped by explicit rules. In any domain there are many kinds of rules: business rules, fiscal rules, operational production rules; contractual rules, safety rules, cultural norms, an so on. But there has been no common, systematic and efficient way to communicate obligation, permission or encouragement from rule-makers to rule-takers. This would facilitate consistency, reliability, and efficiency of rule-based decision-making throughout value-chains.

1 Immediately prior to the formal defense of the dissertation, I researched and prepared a brief review of the most recent academic literature concerning how informatics has been changing the project management domain. Rather than adjusting this introductory chapter, I have added: *Appendix E: Recent Literature on Informatics and Project Management (Afterword, December 2022)* The "gap in project management theory" which existed when I began this research is now smaller, and this thesis is an additional contribution towards filling it. This design research is oriented to further narrowing the gap in project management practice that has persisted. It is the currently only available design for "providing a decentralized distributed data processing pipeline that could enable anyone to publish, discover, fetch, scrutinize, prioritize and have the capability to automate rules on any informatics network."

Normative data expressing MUST, MAY and SHOULD, and their various negatives and synonyms is structured in this system as a distinct class of data with intrinsic directionality that can be instantaneously discovered and transmitted over the Internet, in a form that is directly usable by non-specialized humans and machines, for any purpose, in any language. This offers a general-purpose solution to the problem of communicating which way is 'forward' when orienting decisions involving many people, in on-going micro-level decisions for the day-to-day management of projects, programs, portfolios or platforms, and in core macro-level system design initiatives, mechanisms and structures.

Our concern here is with the design of a method to afford any person or organization the ability to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate rules across any informatics network, with precision, simplicity, scale, speed and resilience, along with deference to prerogatives, agreements and preferences.

Outside the academic realm, this document performs a concurrent duty as a versioned specification for beta system builders. It further serves an operations planning manual for teams of people in public and private sector organizations who are actively considering deployment, and require management understanding of how it is intended to work.

The objective of the research leading to this dissertation has been to conceptualize, design and demonstrate the operational feasibility of a decentralized distributed data processing pipeline that could enable anyone to publish, discover, fetch, scrutinize, prioritize and have the capability to automate rules on any informatics network with precision, simplicity, scale, speed and resilience, along with deference to prerogatives, agreements and preferences. A method is described to improve the communication of rules by providing a way to express them as a distinct class of *data with direction* that can be instantaneously discovered and transmitted over the Internet, in a form that is directly usable by non-specialized humans and machines, for any purpose, in any language.

Until now there has been no common efficient way to communicate obligation, permission or encouragement from *rule-maker agents* to *rule-taker agents*. When using such terms here, no assumption is implied about power relationships. In various contexts, human agency is oriented in hierarchical-tree, hub-and-spoke and peer-to-peer structures. Any individual or entity might like or dislike a rule; advocate or challenge it; conform with or evade it; understand or be confused by it. Each rule-taker agent, be they individual or an entity, operating directly or through a machine, ultimately exercises their discretion about whether or not, or to what degree, to act in accordance with any rule-maker's assertion. (Conrad, 1988) However to make such a decision, they need to know about the existence of such a rule, the *de jure* authority of the rule-maker, the rule-maker's degree of commitment towards fulfillment, and the gravity of non-compliance. (Verhulst et al., 2013).

The approach adopted here interprets rule systems through a theory of agency in a micro-meso-macro framework described by Kurt Dopfer, John Foster, and Jason Potts (Dopfer et al., 2004), together with the body of Project Management (PM) theory at “the interface between projects and the organizations, communities, and networks in and through which projects operate”. Austrian economic geographer Gernot Grabher refers to this as “project ecology” (Grabher, 2004) (Grabher & Ibert, 2011) . By “agency” we mean the possession of attitudinal, intellectual and tangible faculty of action to pursue a specified result.

The contribution of the present research in relation to the PM discipline is to solve the fundamental yet generally overlooked basic requirement of directionally orienting decisions involving many stakeholders in on-going micro-level day-to-day management of projects, programs, portfolios or platforms, as well as in core macro-level system design initiatives, mechanisms and structures. Within a project team, for inter-project portfolios and for value chains interactions, any given agent may or may not be aware of certain rules, but would prefer to be notified about them. And the issuers of the rules may or may not know about any particular agent, but prefer to have a practical way of communicating with them. This research bridges the conceptual and operational gap between rule-makers and rule-takers. It requires only the working premise that agents would communicate out limited data about their identity, task, scenario and context in order to discover what rules are *in effect for,* and *applicable to*, and *invoked by* those facts.

In pursuit of an empirical demonstration of the proposed conceptual foundations and theoretical model, the methodology includes the design of operational specifications, the development of those iteratively emerging specifications into working software, and active peer review of incremental versions of that software by working professionals in various domains.

1.2 Problem Statement

A project management team must take into account rules expressed in contracts, agreements, legislation, regulations, case law, advisories, directives, standards, manuals, protocols, principles, guidelines and informal conventions. Some rules must be complied with fully, some on a ‘best effort’ basis, and others are optional.[2] Rules get updated from time to time, but on no particular schedule. Each organization exists within a municipal, regional and national jurisdiction, all of

2 The representative general framework for compliance assurance throughout both academia and industry is reflected in the ISO 19600:2014 guideline on compliance management systems (ISO, 2014). This guideline is relatively new and is not yet very widely referenced, but it rests upon the long-established and widely used ISO 9000 and 14000 series management quality standards. ISO 19600:2014 offers structured "guidance for establishing, developing, implementing, evaluating, maintaining and improving an effective and responsive compliance management system" applicable to all types of organizations, adjustable for their size, structure, nature and complexity, based on principles of good governance, proportionality, transparency and sustainability. However ISO 19600:2014 does not offer any guidance for the methods by which managers may discover and obtain all the rules they are to comply with.

which issue and enforce their own rules. Various bilateral and multilateral agreements may be in force. The contracts, logistics and entities that every organization is involved with commonly from, or they span, other jurisdictions with different rules. This is a particular domain within the general problem stated in 1948 by information engineer Claude Shannon, to which our present context is added within the square brackets:

> "The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message [in our context, a rule] selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication [about the meanings of rules] are irrelevant to the engineering problem [of designing a rule transmission system]. The significant aspect is that the actual message [containing the correct rules for any particular circumstance] is one selected from a set of possible messages. The [rules] system must be designed to operate for each possible selection [from the entire repository of rules-as-data], not just the one which will actually be chosen since this is unknown at the time of design." (Shannon, 1948, p. 379, fig. 381)

Project managers need to be able to readily disseminate, discover and obtain rules that are 'in effect' for the given dates/times and prerogatives relating to identities and jurisdictions they are concerned with; that are 'applicable' to the class of endeavour and task they are undertaking; and that are 'invoked' by the particular circumstance of the moment. The rule could be fiscal and regulatory instruments of governance; contracts and financial systems in markets for goods, services, assets and currencies; or control systems for networked machines and physical or information infrastructures. When managers cannot readily access such rules at the moment they're needed, they can fall short on compliance and due diligence practices, and they can miss taking advantage of beneficial opportunities, resulting in eroded efficiency and effectiveness. (Suprapto et al., 2015)

The problem has heretofore has been overlooked in the formal literature of this discipline, although it receives attention within industry because inefficiency and poor performance constitute entrepreneurial opportunities for creative firms to design and bring to market work-around methods, partial solutions, and full solutions (Dean & McMullen, 2002) (Driouchi & Bennett, 2012). For example an entire industry sub-sector of competing firms has arisen for the purpose of automating transaction taxes, exemptions, credits, and import/export duties, even though all of the relevant public sector fiscal authorities could have implemented these rules in a generic automated way from the outset. Several companies offer global commercial solutions for the automation of taxes, exemptions and credits. Numerous additional firms provide fiscal rule automation services within regional markets around the world. And there are similar automation commercial automation services for every other fiscal rule set from government.

However, in order for rules to be operational amongst all these automation systems, each rule and then each rule update issued by an authority must be monitored by subject matter experts, who are often lawyers, and maintained by software developers in the various programming languages of their respective automation systems, then tested prior to deployment, and documented in their respective manuals. This redundancy is costly. For example, it is common for commercial fiscal automation services to retain 2% to 8% of the taxes they collect on behalf of government. (Browning, 2013)

Such commercial automation services can be worthwhile in helping organizations at the micro-level to reduce the inefficiency of compliance and reducing ineffectiveness due to compliance failure. But at the meso-level (rule systems) of design, this *ad hoc* arrangement seems sub-optimal.

This design research provides a rationale, a functional specification and partial prototype working components to solve the following general class of problem:

> *Agent A, interacting with Agent B, requires knowledge of one or more externally-managed rules from Agents C..n that are 'in effect' for given contexts, and are 'applicable' to a set of event categories, and are 'invoked' by particular circumstances, where:*
>
> *(i) A and B may or may not know about C..n's rules, or about any updates to them, but either or both would prefer to obtain all available facts about relevant rules when interacting.*
>
> *(ii) C..n may or may not know about A and B in particular, nor about their particular medium of interaction, but can expect A or B or their medium of interaction to be capable of exchanging data with a generic medium common to A..n.*
>
> *(iii) A and B would tolerate the risk of exposing limited data through the generic medium so that it can be used to select information about relevant rules from C..n.*

A general-purpose solution to this problem is a true "Internet of Rules" worthy of the name – a method by which independent, self-contained rules are transmitted efficiently and flexibly from the source repositories in which they are maintained, to the applications that use them. This proposed concept is adapted from the Internet Engineering Task Force (IETF) document "Architectural Principles of the Internet". There one finds the following succinct functional meaning: *"The network's job is to transmit datagrams [independent, self-contained messages] as efficiently and flexibly as possible. Everything else should be done at the fringes. ... Fortunately, nobody owns the Internet, there is no centralized control, and nobody can turn it off. Its evolution depends on rough consensus about technical proposals, and on running code."* (Carpenter, 1996) It is acknowledged, nevertheless, that William Lehr, David Clark et.al. emphasize that there cannot be a single definition of the Internet. (Lehr et al., 2019)

One may suppose that A, B and C are project managers in the sense defined by Gaddis. A project manager (A or B) should be able to easily discover and obtain all third party rules (from C) that are 'in effect' for given dates/times, identities and jurisdictions; are 'applicable' to an undertaking; and are 'invoked' by a circumstance. From the other direction, any project manager (C) issuing specification-conformant rules should be able to have those rules discovered and transmitted automatically on-demand to all parties (A or B) who are in the midst of transactions to which rules may be applicable, without specific knowledge about those other parties and their activities.

In addition to this functional problem to solve, there's an aesthetic aspect of this work which is intrinsic to any design undertaking (Bloch, 1995) , which is put forward as a set of explicit design virtues and design norms. These are expressions of intent. Actual outcomes through implementation in the real world in coming years will establish whether or not this qualitative pursuit achieves the designer's intent.

1.3 Structure of This Dissertation

This dissertation reports on a trajectory from the description of a problem, the descent to bedrock theory in domains such as the philosophy of rules, linguistics, data science, and logic programming, through the climb back up again via design and development of a general-purpose solution to a class of problem.

This work steps through the essential concepts, feasibility, generalizability and utility of a networked computational method for anyone to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate normative data relating what 'is' with what 'ought' to be, based on authority, agreement or preference. The research methodology is grounded in an iterative procedure described in the academic 'design science' body of literature. The fundamental nature of 'rules' in human social and institutional life are made relevant to the methodology by including a set of essential design virtues and norms. Rules and rule systems comprise the 'meso' level of the micro-meso-macro 'project ecology' framework. (Dopfer et al., 2004) (Grabher & Ibert, 2011) The methodology is followed by an elaboration of several core concepts that shape the overall approach.

With the theoretical foundations in place, the purpose of this thesis is elaborated as a comprehensive design rationale and detailed operational description of a new, general purpose type of decentralized, distributed network service. This took some unexpected turns:

- The data model and computational methods required to solve the initial problem have required the re-discovery and re-formulation of tabular declarative computing methods, which have not seen much use for more than a quarter century.
- The data structure and computing algorithm for logic gates would be most workable with a tetranary (four-valued, four-element) vertically-arranged Input/Output table ("vertical I/O table"), versus the more widely known binary 'truth table' method.
- A narrowly-constrained syntactic structure for the sentences expressing the rules would enhance rather than limit usability across semantic domains, and enable full multi-lingual expressiveness.

This dissertation wraps up with six substantive appendices: a unique way of thinking about how to structure decentralized, distributed informatics systems; a set of excerpts form industry literature about my DWDS, or about some of the about use cases it would enable; detailed comments by three independent volunteer contributors to the first reference implementation of the DWDS design; an "Afterword' prepared in December 2022 to review the most recent academic literature about the influence of informatics on project management; and in closing, excepts from a genuine submission to a regulatory body with suggested path for deployment.

When designing something novel, one needs to: (a) explain how it would work; (b) clarify the principles that enable it to work; (c) make the case why others would want it to work; and (d) find people and organizations with the wherewithal to jointly commit towards making it work. Then, one has to (e) actually make it work! This dissertation completes the first two of these tasks to provide the results of theoretical and applied design research, and addresses the third task in a generic context. The fourth and fifth tasks towards implementation have been undertaken as concurrent iterative experimentation as required to improved the first two.

Chapter 2: Methodology

2.1 *Purpose* and Methodology of a DBA versus a PhD

The purpose of a DBA (*doctorate of business administratio*n) degree is unlike that of a PhD (*philosophiae doctor*) degree in the sense that, generally speaking, a PhD candidate pursues original research of academic value towards advancing theoretical understanding, whereas a DBA candidate applies available theory in an original way to solve a general class of real-world problem in professional practice.

This work is rooted in the 'pragmaticist' scientific style of William James who wrote "if you follow the pragmatic method ... [t]heories become instruments, not answers to enigmas" (James, 1922, p. 53). Recognizing that formal science encompasses a wider range of primary reasoning structures than Karl Popper's scientific method, as emphasized by Paul Feyerabend (Feyerabend, 1982) (Feyerabend, 1993) (Feyerabend, 2011) and Nicholas Maxwell (Maxwell, 1972), this work is situated in a the pragmatist school of thought chronicled by James Kloppenberg (Kloppenberg, 1996), Robert Brandom (Brandom, 2008) and Pierre-Luc Lalonde et al. (Lalonde et al., 2010) Sandra Rosenthal and Patrick Bourgeois summarize pragmaticism as follows:

> "[W]hen one turns to the history of modern science rather than to its assertions, what results is a rejection of the 'passive spectator view of knowledge and an introduction of ... the active, creative agent who - through meanings helps structure the objects of knowledge and who thus cannot be separated from the world which he knows. ... In brief, scientific method represents a self-corrective rather than a building-block model of knowledge. (Rosenthal & Bourgeois, 1977, p. 57, 59)

This 'pragmaticist' tradition was captured by David Clark, the first chair of the Internet Architecture Board in the adage: "rough consensus and running code". (Clark, 2018) (Russell, 2006) When bringing forth new designs, practical agents do not necessarily stick with the 'tried and true'. They have an attitude and methodology to step incrementally into the unknown. In *A Treatise on Probability* published 100 years ago, John Maynard Keynes distinguished between the part of our rational belief which is based upon empirical observation and inference, and the part which is based on deductive reasoning through logical reflection (Keynes, 1921). He explained the process by which we establish how an argument about what to expect in the face of uncertainty can be accepted as valid *when empirical proof cannot be obtained*. This is at the core of exploratory design research. A pragmatic agent uses deductive reasoning to step through a series of premises and reasoned conclusions, so that the eventual joint coherence of the series of logical propositions provides the foundation for decisions. (Keynes, 1921, p. 9)

> “Given the body of premises which our subjective powers and circumstances supply to us, and given the kinds of logical relations, upon which arguments can be based and which we have the capacity to perceive, the conclusions, which it is rational for us to draw, stand to these premises in an objective and wholly logical relation. Our logic is concerned with drawing conclusions by a series of steps of certain specified kinds from a limited body of premises”. (Keynes, 1921, p. 17-18).

The present DBA dissertation describes the practical design research that has been undertaken to solve *a class of real-world problems*. Instead of being pursued as a ‘theoretical research’ PhD dissertation in the department of a traditional specialized ‘discipline’ to resolve conceptual problems, this is a ‘design research’ DBA being pursued through a multi-disciplinary academic program offering a doctorate in administration - project management (‘Doctorat en administration - gestion de projet’, Université du Québec en Outaouais). This particular academic program is founded essentially upon the same pragmaticist orientation discussed above: “contributing to the advancement of knowledge focused more on the application of project management knowledge than on the development of theory.” (UQO, 2011) [3]

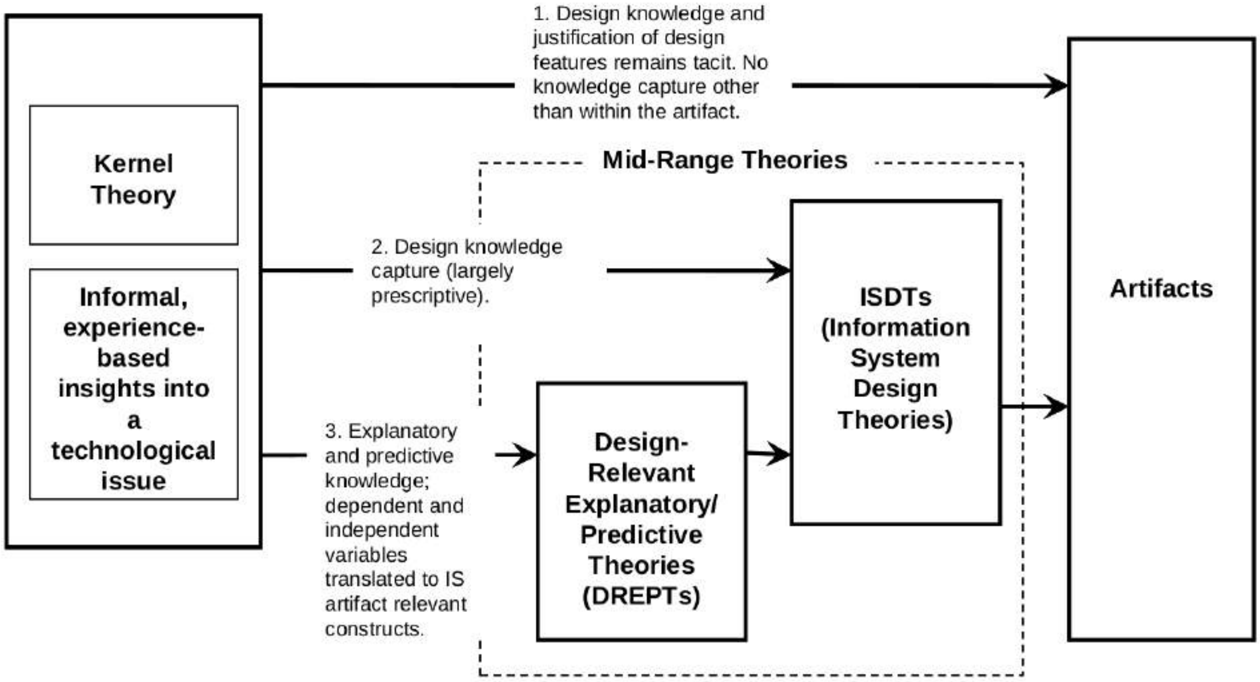
2.2 Methodology of Middle Range Theory

A conceptual map for understanding ‘middle range theory’ is reproduced in Figure 5, from a paper by William Kuechler and Vijay Vaisshnavi, entitled: “A Framework for Theory Development in Design Science Research” (Kuechler & Vaishnavi, 2012). Their work arises from information systems design theory (ISDT), which is directly suited to the present dissertation research, but also seems readily generalizable to any domain. Their framework is elaborated in their book by the premier technical publisher, CRC Press: “Design Science Research Methods and Patterns: Innovating Information and Communication Technology” (Vaishnavi & Kuechler, 2015).

They explain that: “an ISDT [information systems design theory] is by its nature and intent, prescriptive. An ISDT is similar to what is called a model in computer science and some engineering disciplines ... [I]t provides high level definition of the functioning of an artifact to achieve a design goal and direction toward its construction, but does not describe how the artifact works or by what mechanism(s) the meta requirements and design method achieve the design

3 Although some faculty in this department prioritize contributing to a theoretical discipline *about* project management, when the program was created in 2011 its charter description explained: “Il se distingue également d’un programme de Ph.D. par le type de recherches que les étudiants seront appelés à entreprendre; celles-ci viseront en effet à contribuer à l’avancement de connaissances centrées davantage sur l’application du savoir en gestion de projet que sur le développement de la théorie. (UQO, 2011) (Translation: “It is also distinguished from a Ph.D. program by the type of research that students will be called upon to undertake; these will be aimed at contributing to the advancement of knowledge focused more on the application of project management knowledge than on the development of theory.”

Figure 1: Middle-range theoretical design research is distinguished from kernel theory development and empirical theory. (Kuechler & Vaishnavi, 2012)



goal. ... rather than codifying the design for a specific artifact implementation, an ISDT captures meta-requirements and a meta-design that is applicable to a class of artifacts." Typically in a DSRIS [design science research in information systems] effort, "the kernel theories are taken from the most current literature of another field." (Kuechler & Vaishnavi, 2012, p. 411) (Kaufmann, 1985) The philosopher of research methodology, Nancy Cartwright, emphasized recently: "middle-range theory is about things that come under the label 'mechanism' " (Cartwright, 2020, p. 269).

The present work is undertaken in parallel with testable partial implementations among independent professionals, a *de facto* type of concurrent engineering (Maranzana et al., 2008). By making early drafts of this dissertation available through free/libre/open licensing and relationships the work has benefited from independent 'professional peer review' germane to a pragmaticist DBA orientation towards solving real-world problems. A public sector data scientist and a private sector software designer proceeded to implementing version 3.x of the DWDS design as it was documented in the December 2021 draft of my dissertation, through their own respective initiative and effort, outside of any contract or compensation. This provides genuine validation. Granted, some implementation work is also contracted to current undergraduate students and recent grads. But internally-directed elements do not carry to same weight. In the present applied context, relevant uptake of the publicly-shared research outcomes among working professionals in government and business, and positive reviews by independently-motivated authors in high-quality trade publications, provides tangible evidence of a practical contribution.

Notwithstanding what has just been said, this dissertation contains only very minimal examples of industry or government use cases, limited to where these help to illustrate system design elements. There are three reasons for this constraint:

(a) The details of particular 'real world' projects based upon the newly-created design are out-of-scope for this design research dissertation in order to maintain focus upon the *general purpose* character of the new DWDS design. The methodology has pursued intrinsic *ex ante* generalized design that is intended to be of use in any domain, instead of *ex post* generalization from one or more initial use-cases in particular domains. The elaboration of specific or archetypal use-cases would distract from, rather than highlight the many carefully deliberated design choices that have been made to attain general-purpose utility. My prior experience is that an emphasis on case studies and examples tends to limit an audience's perceived range of applicability.

(b) Most of the real-world implementation and collaboration projects that have already arisen from the present design research are organized and led by people and organizations with no prior connection to the present author. It is precisely because they are initiated and managed independently of my control that they carry weight as evidence of the general purpose utility of this design contribution. Accordingly, Appendix C contains excerpts of what others have published that acknowledge use of the present research contributions.

(c) Genuine implementation projects in which I personally use the outputs of this design research, and which internally motivate me to pursue it, are left *outside* the scope of this dissertation. This is because I am a biased user. To demonstrate that my design works in projects that I control would be subject to the well-known *"It works for me!"* fallacy. Every such implementation is inevitably framed in ones' own "mental constructions and concepts", and merely "confirms his or her constructed view of reality". (Shindler, 2010, 347-348) Interested readers can locate my other initiatives easily enough online. The design contribution described herein must stand on its own merits, and attract users independently of my own implementation interests.

2.3 Design Research

Both positivist and constructivist views are engaged in model-based design research (Jonas, 2006) (Horváth, 2007) (Faste & Faste, 2012), situated in an action theory of projects (Bredillet, 2005). The positivist elements of this research work are subject to constructivist influences of the researcher as an active agent (Archer, 2007), in a process that Sinkovics and Alfoldi refer to as "non-linear progressive focusing" (Sinkovics & Alfoldi, 2012). Kingston, Henderson and Vernik have succinctly illustrated (Figure 2) three primary requirements of a good research problem: that it is perceived by stakeholders to be a problem to be solved; that it is indeed solvable; and that its resolution worth pursuing. (Kingston et al., 1999)

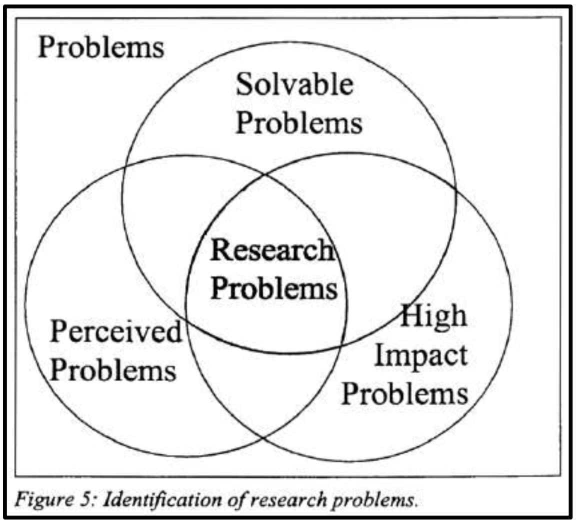


Figure 2: Criteria for a Research Problem (Kingston et al., 1999)

These lead pragmatically to three essential hypotheses, none of which can be taken for granted for a particular problem:

- H1: The problem is perceived by stakeholders.
- H2: The problem is solvable.
- H3: At least one solution to the problem can be developed and deployed with high impact.

All three hypotheses are relevant to a constructive design research project that would enable anyone to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate rules on any informatics network. The third hypothesis, as it is stated here, actually incorporates two separate hypotheses : that at least one solution can be developed, and that it can be deployed with high impact. To determine whether a solution can be deployed also requires qualitative analysis of the generalizability and utility of the design. All elements of this set of hypotheses benefit from historical and conceptual research methods in order to track down sources of ideas upon which to establish an informed research design initiative.

Four channels were engaged in the course of this research:

- Intensive investigation and study of primary and secondary academic, industry and historical sources across a wide diversity of complementary domains.
- Proactive sharing and discussion with others of my on-going research through free/libre/open licensing and venues, an online project management application, social media and conferences.
- Structured collaboration via the not-for-profit Xalgorithms Foundation, which I incorporated to manage research funding, issue contracts for supporting work, and host working groups.[4]
- Reflections on my own parallel involvements in various projects involving industry, government and other organizations.

This project is structured in the tradition of "design science" which Vijay Vaishnavi and William Kuechler describe as "research that uses artifact design and construction (learning through building) to generate new knowledge and insights into a class of problems" (Vaishnavi & Kuechler, 2015, p. 396) (Chakrabarti & Lindemann, 2015) (Wieringa, 2014) The process employed here is described by Rudolf Sinkovics and Eva Alfoldi as "non-linear progressive focusing" (Sinkovics & Alfoldi, 2012) undertaken through multiple iterative cycles as in Imre Horváth's "design inclusive research" methodology (Horváth, 2007):

(a) Observe, describe and reflect on the current state of knowledge and application;

(b) Invent concepts, models, methodologies and designs, grounded in theory and practice;

(c) Assess validity and feasibility of designs through building and testing real instantiations.

This project is also shaped by a particular design research style:

(d) Identify and align to elemental design virtues and design norms, in the manner emphasized by Tim Berners-Lee in "*Principles of Design*" (Berners-Lee, 1998a);

(e) Implement working prototypes in the pragmatic tradition of the Internet developer community (Russell, 2006), which reflects "concurrent engineering" as defined by Nicolas Maranzana and Emmanuel Caillaud (Maranzana et al., 2008) and "learning consortia" explained in the work of Edgar Schein (Schein, 1995);

(f) Pursue "engaged scholarship" relating know-how of industry practitioners with the work of concept theorists, following the guidance of Andrew Van de Ven and Paul Johnson (Van De Ven & Johnson, 2006).

(g) Excavate root sources in the evolution of knowledge in the manner encouraged by (Popper, 1979, pp. 238–239), which can lead to the resurfacing of useful ideas and techniques that have been overlooked or forgotten.

4 As of the date of thesis submission, Xalgorithms has under paid contract for supporting work two undergraduate students (Northwestern), two graduate students (Stanford, Northwestern) and three recent graduates (Simon Fraser, Ottawa, Maryland Institute).

Three interacting layers are succinctly described by Elizabeth Steiner: "models can be devised from theory so that theories can impact upon practical decision-making (Steiner, 1998, p.9-10, emphasis added)". These layers show up independently in Imre Horváth's phases of design inclusive research as shown in Figure 3.

- *Theory/Conception:* Pre-study conceptualization research to determine how to examine the research problem;
- *Model/Design:* Creative design research on models, and methodologies to creating a testable "proof of concept" instantiations;
- *Decisions/Prototyping:* Validation of the research methods and design methods and the artifacts developed, with regard to generating useful domain knowledge.

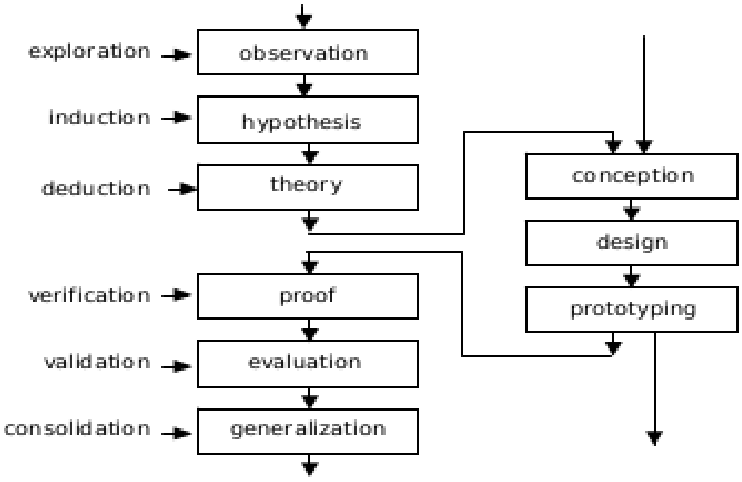


Figure 3: Phases of design-inclusive research (Horváth, 2007, p.6)

In this structure, the System Development Life Cycle (SDLC) involves rapid, informal prototyping of partial, short-term visioning mock-ups, trial-and-error innovation; along with low-level problem-solving and bug fixing. It is sometimes referred to as '*code and fix prototyping and visioning*', which is analogous to improvisational jazz. It involves a process of continually identifying what is working and what is not working, and making the necessary corrections. Facilitative licensing terms and conditions enable rapid knowledge-sharing and learning relationships with others. This iterative approach is optimally suited to exploratory software systems design research and development. If the design eventually achieves core stability, the team can shift to a more structured methodology to systematically advance upon that design.

The particular methodological juxtaposition of a deep theoretical pursuit and functional application experimentation brings a special synergy to this undertaking. Jawed Syed, John Mingers and Peter Murray explore how the divergent perspectives of scholars and practitioners can be bridged with a critical realist approach (Syed et al., 2010). And John Reed, former CEO of Citigroup has emphasized that "maintaining the balance between core research and applied research is as important to us as it is to the academic enterprise" (Huff, 2000, p. 62).

Pierre-Luc Lalonde, Mario Bourgault and Alain Findeli list four types of theory-practice relationships in project management literature:

i. Detached prescriptive theory for guiding practice;
ii. Detached interpretive theory for understand practice; and
iii. Engaged reflective theory-in-action, and
iv. Engaged intuitive non-theoretical practice (Lalonde, Bourgault, & Findeli, 2010, p. 31)

Project design research is the third type: reflective theory-in-action, a type of *engaged scholarship* where the researcher co-produces with practitioners both academic knowledge and practical know-how (Van De Ven & Johnson, 2006). Kari Kukka and Petri Suomala refer to this as "interventionist research" in business (Lukka & Suomala, 2014). Engaged scholarship spans professional and scientific orientations (Ormerod, 2009, p. 1209). This involves a commitment to bind conceptual research with tangible problems in all of their implementation complexity.[5] Although such "design research" can be considered interventionist, it differs from "action research" in several ways, as reflected in Table 1 produced by Antonio Dias de Figueiredo (Dias de Figueiredo, 2018) Deployment of the reference implementation software and of the network service to project managers is beyond the scope of the present *design research*. While this work is intended to eventually change the way project managers operate, and although it is attracting some interest among project managers 'in the field', there can be no *action research* analysis and reporting involving project managers prior to completion of the present system specification.

5 Two thousand years ago Aristotle characterized three essential types of human activity:

- *Theoria* as the activity of which the purpose is to understand;
- *Praxis* as activity of which the purpose is the action itself; and
- *Poiesis* as activity of which the purpose is production. (Aristotle, 1991)

Humberto Maturana and Francisco Varela have distinguished two forms of *poiesis*. They coined the term *autopoiesis* to refer to the activity of an agent specifying and re-producing itself; and *allopoiesis* to refer to the activity of an agent producing a class of thing distinct from itself. *(Maturana & Varela, 1980)* So we may say that the present research is in the realm of *theoria-allopoiesis*.

Table 1: Comparison of "Action Research" and "Design Research" methodologies. (Dias de Figueiredo, 2018)

	Action Research	Design Research
Aim	understanding a problematic situation and trying to solve or improve it	creating an artifact (object, software, framework, strategy, theory) and improving it by using it
Object	a problematic situation	an artifact (object, software, framework, strategy, theory)
Unit of Analysis	the context where the problematic situation occurs	the artifact to be created and its environment
Data Collection	all standard qualitative research approaches	all standard qualitative research approaches
Core Concepts	planning, acting, reflecting, cycles, participatory action research	design, design-based research, artifact
Report	description of the problematic situation, its solution and the knowledge developed	description of the artifact and of the knowledge gained by developing it

Collaborative learning with a community of scholars and practitioners exposes the researcher to diverse perspectives, alternative ways to frame problems, as well as alternative concepts and models (Sinkovics & Alfoldi, 2012).

Some short segments of “Section 4.2 Available Methods for Logic Data Models” were also checked durng the final edit of this dissertation using the beta version of OpenAI/GPT-3, a recently-released dialogue service based on language patterns. (OpenAI Inc., 2021) The OpenAI policy on attribution for the resulting arguments and expressions is as follows:

> “***Content co-authored with the OpenAI API policy:*** Creators who wish to publish their first-party written content ... created in part with the OpenAI API are permitted to do so under the following conditions: The published content is attributed to your name or company. The role of AI in formulating the content is clearly disclosed in a way that no reader could possibly miss, and that a typical reader would find sufficiently easy to understand.” (OpenAI, 2022)

The segments of the final text within Section 4.2 that I reviewed through this natural language online dialogue are those in which I compare and contrast various systems of multi-valued logic and computation. These segments are indicated with the following citation: (Potvin & OpenAI/GPT-3, 2022) The results of that computer-human interaction required thorough correction and refinement by the present author. Yet some useful substantive distinctions are attributable. The unedited transcripts of these text conversations have been saved, and are available upon request.

No human ‘subjects’ other than research colleagues and project implementation stakeholders have been involved in the dissertation research. The methodology has not involved the collection of and non-public, personal private, or commercial confidential data. No interview or survey data has been assembled to be analyzed. My interactions with other academic and professional researchers and systems designers has been via publicly accessible communications, mainly though weekly consultative video meetings that are promoted on the project website *https://xalgorithms.org/* These discussions are always open to the public. There generally occurs a balanced round-table sharing of our respective research, development and outreach ideas. Participatory implementation of the component software is invited through free/libre licensing and early sharing of the DWDS design specification as a work-in-progress. Under terms of Creative Commons Attribution License 4.0 International, the Apache 2.0 License, and the Affero GPL License, there have been three types of relationships between implementers and the present doctoral candidate. Various professionals and academics found my interim results helpful towards advancing their own applied research, projects and publishing. Also I have hired several undergraduate students and recent grads to advance particular implementation work.

2.4 Design Success Criteria

The target outcome of this work is to have a rationale, a functional specification and partial prototype working components to solve the following general class of problem: “*Agent A, interacting with Agent B, requires knowledge of one or more externally-managed rules from Agents C..n that are ‘in effect’ for given contexts, and are ‘applicable’ to a set of event categories, and are ‘invoked’ by particular circumstances”* amid uncertainty about agents and about rules.

The success of this design research can be assessed in relation to three criteria:

Is it plausible that the target system, once implemented, would be capable of enabling anyone to express, publish, find and fetch rules:

1. *At scale over the Internet? (Feasibility)*
2. *Across any rule domains and use cases? (Generalizability)*
3. *More effectively (greater outcomes) and/or more efficiently (less time/money/risk) (Utility)?*

These criteria cannot be applied directly to *a meso-level target outcome*,[6] because that would require a full external implementation or deployment of the end-to-end service. Even though this design research is driven by an overall motivation to improve rules and practices across whole industries and markets, the criteria can only be applied to *the rationale, the functional specification, and the partial implementation of the working components.*

The partial building and testing of novel elements of the target system, undertaken iteratively as part of the design research process, helps the designer and participating contributors to think through the coherence of each part, and the end-to-end composability of the whole. Implementation and deployment of the design as a fully-working service would be a whole team undertaking involving several types of specialists, with many requirements beyond the scope of the specification itself. For example, genuine implementation would involve development and testing of relatively sophisticated software components, design and set-up of a user interface, the preparation, documentation and adherence to a security model, preparation of administrator guidance on system deployment, maintenance and adaptations, a set of organizational and financial arrangements, and a community engagement process.

Notwithstanding these boundaries of validation, in fact throughout the first half of 2022 there has been considerable amount of external work underway to create an end-to-end reference "version 3.x" implementation. This has been pursued independently by free/libre community participants on a *pro bono* basis, who otherwise work in commercial and public sector organizations. In order to obtain many precise user-interface details which would be more time consuming than an otherwise fully-employed volunteer could attend to, I hired an undergraduate student to work full-time during the summer of 2022. All of the autonomous version 3.x implementation detail is available under free/libre licensing, and technical progress is documented in sections 6.2.3 and 6.3.3 of this dissertation.

Validation of the newly-structured method of rule expression by transcribing a significant body of real-world legislation, agreements and contracts would similarly require a greater amount of work than is appropriate at this stage. This is a specialized undertaking to be done with subject matter specialists. Within the research methodology several individual clauses or sections from laws, policies and agreements have been tested successfully by the present author. That effort itself actually involves, for the most part, deconstructing complicated source rules in natural language into their constituent simple sentences in natural language. This is not really a test of the system. The relevant question really is: Can the newly-designed method scale in real settings? Only a semi-independent

6 The intended *meso-micro-macro* sequence in society is that an improved rules system (meso-level) that is able to enhance performance of decision-makers within organizations (micro-level), will tend to proliferate, and upon reaching critical mass, can result in transformative effects for whole societies and economies (macro-level). Obviously such outcomes cannot be evaluated directly or within an immediate time scale of a DBA, and it is unlikely that any simulation meso-micro-macro model of society, or any selection of interviewees commenting on their expectations of the implementation this system in society, would be free of actual or perceived confirmation bias.

initiative *at scale* would provide genuine validation of the new method of expression. Presently, a small selection of tested rule samples provides a reasonably credible demonstration.

Generalizability is pursued intrinsically by working from *ex ante* principles and virtues described in Section 2.5, instead of extrapolating from the solving of various particular problems towards *ex post* generalizability.

2.5 Design Virtues and Norms

An explicit set of *design virtues* which are grounded in ethics, and *design norms* based in pragmatism, have guided our systems design research towards the DWDS, and have enabled consistency of purpose throughout its project trajectory.

Twenty years ago in "*Principles of Design*" Tim Berners-Lee encouraged the general criteria of: 'simplicity', 'tolerance', 'modularity', 'decentralization', 'interoperability' and 'least power'. (Berners-Lee, 1998a). For making purposeful choices and for evaluating outcomes of the design research and the operational DWDS, we augment his list with four chosen criteria: 'free/libre/open relationships', 'intuitiveness', 'human-centred automation', and a 'tabular declarative style'.

Following is our perspective on each of these qualitative design virtues and norms which should be observable in the result. Our list seems to be generally consistent with the *"Ten Theses on Logic Languages for the Semantic Web"* described by François Bry and Massimo Marchiori (Bry & Marchiori, 2005), but our scope concerns a specialized end-to-end system design that has additional requirements.

2.5.1 Design Virtues

2.5.1.1 Human-Centred Automation

In her pivotal work *Ironies of Automation*, Lisanne Bainbridge explained why and how "designer errors can be a major source of operating problems" (Bainbridge, 1982, p. 129), and she provided the kernel of understanding to 'human-centred automation' (Graeber & Billings, 1989) (Mitchell, 1996) (Nadeem, 2019) (Strauch, 2018) (Neyland & Möllers, 2017) An automated system may incorporate a conceptual or implementation error, could be receiving faulty data (Ethiopia, 2019) (Indonesia, 2018) or be compromised by an adversarial attack (Finlayson et al., 2019) (Knight, 2019) (Tencent, 2019). Less ominously, there are more and less efficient ways to accomplish a purpose, and the automation designer and operations manager might have different preferences or priorities.

Human-centred design in the general domain of computational rules systems would never automate the *imperative* imposition or enforcement of rules. Instead, each person who is subject to a rule ultimately retains their inalienable prerogative of discretion about whether or not, and to what degree, to act in accordance with it (D. A. Conrad, 1988). Without such a premise of agency, prerogative would be superfluous.

Kees van Dongen and Peter-Paul van Maanen refer to empathy, accountability and error management as three ways in which human-centred automation designers may assure end-user agency (van Dongen & van Maanen, 2013):

Stakeholder Empathy

- Use understandable modular algorithms to aid decisions;
- Ensure that stakeholders know about each decision aid;
- Make the reasoning of each decision aid easily available and understandable;
- Reveal intermediate results in a comprehensible way so users are not lost;

Designer/Supplier Accountability

- Ensure that designers and suppliers of decision aids feel accountable for performance;
- Ensure that designers and suppliers of decision aids are responsible for the outcome quality;
- Ensure that decision aids accommodate of exceptions;

Elegant Error Management

- Anticipate error and incorporate elegant error management into the design;
- Make transparent the potential sources of error, and actual errors;
- Inform stakeholders about conditions in which the decision aid performs well and the conditions in which it does not.

2.5.1.2 Free/Libre/Open Relationships

The 'free/libre/open' way combines active respect for *user freedoms* with the productivity of *open source methods*. These are distinct but complementary demand-side and a supply-side perspectives on the relationships among the designers, operators and users of a system. On the demand side, this is premised upon every user's freedom to run and to use the essential software components for any purpose, study how they work, to adapt them to specific needs, to copy and redistribute them, and to improve them and distribute modified versions. (Free Software Foundation, 1996) On the supply side, those who create, maintain and run software components or derivatives of them chose to authorize unrestricted publication of source code, free redistribution of verbatim or derivative works while respecting the integrity of author's intellectual rights, neutrality regarding technology, fields of endeavour, deployment diversity, and transparency (Open Source Initiative, 1998) This working relationship depends upon both active facilitation and removal of artificial barriers. (Nature, 2014) (Evanko, 2014)

2.5.1.3 Tolerance

Tolerance encompasses a designer's spirit of respect for the prerogatives of those who are users of a design, or who are subject to its result, but also those across the community of other designers who would engage with the designed work in their preferred ways, for their own purposes, within their chosen domains, using their preferred technologies, and interpreted in the contexts of their respective normative paradigms. In practice it means that a designer is attentive to tangible objective biases, and their internal subjective biases. Berners-Lee suggests a notional "tolerance ratio", positioning the degree to which a system designer accommodates users, versus the degree to which users must accommodate the system designer. His illustrative example may seem trivial: "Always say 'http:' in lower case, but in practice understand 'HTTP:' too” (Berners-Lee, 1998b). But this demonstrates experience and understanding relating to potential sources of user frustration. He considers tolerance to be a “guiding rule in internet protocol design”.

This topic merits further reflection. The virtue of tolerance was a central theme in the 1859 book *“On Liberty”* by John Stuart Mill and Harriet Taylor Mill. In particular their second chapter entitled: "On the Liberty of Thought and Discussion" explains how intellectual pluralism is indispensable to solving problems:

> “It still remains to speak of one of the principal causes which make diversity of opinion advantageous, and will continue to do so... We have hitherto considered only two possibilities: that the received opinion may be false, and some other opinion, consequently true; or that, the received opinion being true, a conflict with the opposite error is essential to a clear apprehension and deep feeling of its truth. But there is a commoner case than either of these; when the conflicting doctrines, instead of being one true and the other false, share the truth between them and the non-conforming opinion is needed to supply the remainder of the truth, of which the received doctrine embodies only a part." (Mill & Mill, 1863, p. 88-89)

Tolerance of multiple legitimate perceptions and normative frameworks is well developed in formal logic, and appears across many disciplines.[7] In the realm of applied design, a pluralist demeanour enables the designer to accommodate the full scope of present and potential social scenarios, and provides access to the full range of technical methods.

2.5.1.4 Interoperability

Interoperability is essential to market access and user choice. When nodes of a network are operated arms-length by autonomous competitive organizations, they negotiate common specifications, protocols and component designs through cooperative governance. Some consider this to be ‘collusion’ (OECD, 2016), but others consider it straightforward that some degree of ‘co-opetition’ is indispensable to achieving and maintaining open level market accessibility (Brandenburger & Nalebuff, 1997). Participants in

7 A general philosophy of intellectual pluralism is provided in the often misunderstood and criticized works of Paul Feyerabend (Feyerabend, 1982)(Feyerabend, 1993)(Feyerabend, 2011). Elisabeth Lloyd explains that Feyerabend’s defense of fringe ideas is motivated by “the central importance of a proliferation of views and methods—and the appropriate attitudes of openness and tolerance” grounded in Mills’ work ‘On Liberty’. (Lloyd, 1997)

In the natural sciences, physicists Gregoire Nicolis and Ilya Prigogine explain in their popular book reconciling the emergence of complexity with the unrelenting necessity of entropy, that "a pluralistic view" is generally required to describe the physical world (Nicolis & Prigogine, 1989, p. 5-6). Their "principal message" is that a single mode of perception cannot hope to embrace the whole scope of observed phenomena. Similarly Tim Allen, an ecosystem scientist, has maintained that "several different world views or perceptions are required to solve a problem". (Allen, 1987, p. 25)

In the humanities, anthropologist Michael Smith credited socio-political historian John Sydenham Furnivall as “the first to distinguish the plural society as a separate form of society”. (Smith, 1960, p. 763) In Furnivall’s most general expression, a plural society is one “in which two or more groups live side by side but separately within the same political unit” (Furnivall, 1945, p. 167). In such scenarios demand is heterogeneous but intermingled because “they mix but they do not combine”(Furnivall, 1948, p. 304). Practical wants in the market for goods and services may be similar, say furniture or transportation, but higher-order wants, rules and modes of social interaction, are particular to each section (Furnivall, 1931, p. 178).

a round table session at the 2018 World Trade Symposium considered: “How can silos that prevent interoperability be transformed into nodes of a network?” (Potvin, 2018) Summary results of this discussion are provided in Table 2.

Table 2: “How can silos that prevent interoperability be transformed into nodes of a network?” Summary results of a workshop facilitated by the author at the World Trade Symposium 2018, London, U.K. (Potvin, 2018)

	Expression *How is it stated?*	**Interface** *How is it accessed?*	**Processor** *How does it work?*	**Channel** *What pathway?*	**Infrastructure** *What hardware?*
Ideas & Opportunities	*The payloads for interoperable computation should use language-independent syntax*	*Interfaces should implement standard semantic specifications*	*The system should implement the standard non-repudiation*	*The system design should incorporate mult-path routing.*	*The system design should employ ‘resilience thinking’*
Obstacles, Risks, Problems	*Incumbent methods of expression and semantic models make a transition to interoperability difficult and expensive.*	*Interoperability is a “layered” concept, like layers of an onion, from basic connectivity through to payload semantics.*	*Non-standard data, messages and documents increase and complicate the processing requirements.*	*Some pathway service providers assert a “my platform” attitude and create incompatibilities for competitive reasons.*	*1. Core conforms with standards; 2. Secondary connections meet minimum standards 3. Internal infras-tructure may diverge*
Strategic Context	*Interoperability requires an appropriate legal framework:*	*Assume that any node can be owned and operated by a different organization.*	*Payload standards Transport standards*	*Users choose their own providers who then interconnect.*	*Core infrastructure has cooperative not-for-profit governance. (Members can be profit-driven.)*
Priorities & Sequencing	*1. Legal framework 2. Business model 3. Technical design and implementation.*	*Create a simple modular interface serving basic needs, and add capabilities step-by step.*	*Beyond the initial proof-of-concepts, proceed to design and develop through an international working group.*	*Learn about and build upon the interoperability methods that already exist and extend as needed.*	*Learn about and build upon the interoperability methods that already exist and extend as needed.*

2.5.2 Design Norms

2.5.2.1 Simplicity

Design simplicity is superbly conveyed by aviator and writer Antoine de Saint-Exupéry:

> "Have you ever thought... that all ... industrial efforts, all ... computations and calculations, all the nights spent working over draughts and blueprints, invariably culminate in the production of a thing whose sole and guiding principle is the ultimate principle of simplicity?
>
> It is as if there were a natural law which ordained that to achieve this end, ... there must be the experimentation of several generations of craftsmen.
>
> In anything at all, perfection is finally attained not when there is no longer anything to add, but when there is no longer anything to take away...
>
> It results from this that perfection of invention touches hands with absence of invention, as of that line which the human eye will follow with effortless delight were a line that had not been invented but simply discovered, had in the beginning been hidden by nature and in the end had been found by the engineer." (Saint-Exupéry, 1939, pp. 41–42)

It was aeronautical engineer Clarence Johnson who emphasized "applying the simplest, most straight-forward methods possible to develop and produce new products" and then articulated the famous aphorism: "Keep it simple, stupid—KISS" (Rich, 1995, p 221, 231). System procedures, interfaces, and documentation, can benefit from the well-known 7±2 guideline that average human short-term memory capacity for processing information is constrained to about seven plus or minus two items (Miller, 1994), or its less prominent 4±1 refinement (Cowan, 2001) (Mathy & Feldman, 2012).

Doug McIlroy, a core contributor to the Unix operating system and to the *Unix Philosophy* integrated pragmatic simplicity into his work: the quality, readability and speed of a computer program's source code is improved by decreasing the number of lines of code. He famously said: "The real hero of programming is the one who writes negative code." (McIlroy, 2009) This commonly rediscovered *'less is more'* adage was highlighted recently by Jayesh Srivastava and Li Shu as the "affordance of absence".[8] They describe the addition of utility to a product or service by the elimination of functionality, material or components. (Srivastava & Shu, 2014, p. 7)

David Patterson and David Ditzel addressed this in *"The Case for the Reduced Instruction Set Computer"* (Patterson & Ditzel, 2000) "Some would argue that simplifying an architecture is a backwards step", but they explain why a "Reduced Instruction Set Computer" has four advantages:

- *Implementation Feasibility:* "A complex architecture has less of a chance of being realized in a given technology than does a less complicated architecture"; (p. 28)
- *Design Time:* "[A] design that takes only two years to design and debug can potentially use a much superior technology and hence be more effective than a design that takes four years to design and debug."; (p. 29)
- *Speed:* "If leaving out an instruction or address mode causes the machine to speed up the minor cycle by 10%, then the addition would have to speed up the machine by more than 10% to be cost-effective."; (p.29)
- *Better Use of Chip Area:* "[A]rea gained back ... can be used to make the RISC even more attractive ... For example, … the entire system performance might improve more if silicon area were instead used for on-chip caches, larger and faster transistors, or even pipelining. ... the RISC architecture can always stay one step ahead. (p.29)

They emphasize whole-system performance optimization:

> "By a judicious choice of the proper instruction set and the design of a corresponding architecture, we feel that it should be possible to have a very simple instruction set that can be very fast. This may lead to a substantial net gain in overall program execution speed. This is the concept of the Reduced Instruction Set Computer. ... There are undoubtedly many examples where particular "unique" instructions can greatly improve the speed of a program. Rarely have we seen examples where the same benefits apply to the system as a whole. For a wide variety of computing environments we feel that careful pruning of an instruction set leads to a cost-effective implementation." (p. 31)

In the design of whole systems, the 'elimination' of function from the design can be facilitated through separation of function. McIlroy speaks of disaggregating systems into discrete components: "Make each program do one thing well" (McIlroy et al., 1978, p. 1902) (Raymond, 2003) And Brian Carpenter, *et.al.* explain that the immense general purpose utility of Internet protocols and services is a direct result of pushing all functions other than data transmission away from the relatively simple core, to be handled by the innumerable applications at the edge (Carpenter, 1996).

8 Don Norman brought precision to the term design *'affordance'*, referring to characteristics of a thing which set the realm of possibilities for how anyone may interact with it, independently of *'signifiers'* for any intended interactions (Norman, 2013).

2.5.2.2 Modularity

A designer's commitment to modularity means assuring that each element of a system will be interchangeable with alternative independent implementations. This enables a system specification to be focused on core functionality, while being unbound by implementation details that are not intrinsic to the functional specifications, such as platform, language, infrastructure, services, supplier and other factors. In other words: "Form ever follows function" as expressed by architect Louis H. Sullivan in the late 1800s (Sullivan, 1896). This norm underlies the pragmatic idea of "loose coupling" among discrete components and data sets, so that as the external environment changes, an implementation can be readily adapted in order to maintain the original intended function.

2.5.2.3 Intuitiveness

Christopher Alexander has explained that the most pivotal design insights are typically the most difficult to perceive "because they are so ordinary, that they strike to the core". He observes that: "What makes them hard to find is not that they are unusual, strange or hard to express—but on the contrary that they are so ordinary." (Alexander, 1979, p. 219) Design that is intuitive may seem obvious and uninteresting. We suggest the phrase *Archimedes' Bathos* to refer to this anticlimactic descent from the ideals of design creativity and innovation to the banality of intuitive operation. The Roman biographer Plutarch said of Archimedes, three centuries after his death: "No one else could ever figure out these things on their own. And yet, as soon as they have seen his solutions, they feel that they could of course have worked them out by themselves." (Plutarch, 100 C.E.)

2.5.2.4 Decentralization

In any organizational hierarchy, agency is structured by one of two complementary opposite norms in law (*de jure*) and/or in operations (*de facto*):

- *Centralization:* Agency at any level of aggregation is delegated at the discretion of the relatively more comprehensive decision-makers. In law this is called 'paramountcy'.
- *Decentralization:* Agency at any level of aggregation is delegated at the discretion of the relatively disaggregated or micro-level decision-makers. In law this is called 'subsidiarity'.

Internet architecture, for example, is premised upon shifting application control away from the core, out to the edge (Carpenter, 1996) For computational and communication systems generally, decentralization is facilitated by the separation of logic and control, as Robert Kowalski explains in his article *Algorithm = Logic + Control*:

> "An algorithm can be regarded as consisting of a logic component, which specifies the knowledge to be used in solving problems, and a control component, which determines the problem-solving strategies by means of which that knowledge is used. The logic component determines the meaning of the algorithm whereas the control component only affects its efficiency. The efficiency of an algorithm can often be improved by improving the control component without changing the logic of the algorithm. We argue that computer programs would be more often correct and more easily improved and modified if their logic and control aspects were identified and separated in the program text." (Kowalski, 1979, p. 424)

This is not solely a matter of a suitable technical architecture. In recent years centralized ownership and control of some of the operational elements of the Internet (Internet Society, 2019) has led to what David Clark, founding chair of the Internet Architecture Board (1981-1990) refers to "the ossification of customary business relationships" which he considers to be potentially "more of a barrier to innovation than the ossification of technology". (Clark, 2018, p. 251, 292).

Kinji Mori has defined an "autonomous decentralized system" as a set of self-contained subsystems, each with its own management capability to guide its own operations as well as to coordinate with the others to comprise an integrated system. Subsystems are connected via data-metadata bundles broadcast to and circulated across the network. (Mori, 1984)(Mori, 1993)(Mori, 2007)

2.5.2.5 Least Power

It is commonplace to assume that the richness of expression in a programming language is desirable, but Tim Berners-Lee and Noah Mendelsohn recommend choosing the least powerful language suitable for a given purpose, in order to optimize for data utility:

> Nowadays we have to appreciate the reasons for picking not the most powerful solution but the least powerful. Expressing constraints, relationships and processing instructions in less powerful languages increases the flexibility with which information can be reused: the less powerful the language, the more you can do with the data stored in that language. (Berners-Lee & Mendelsohn, 2006)

Haseed Qureshi explains how constrained expressiveness enhances intrinsic computational security.

> "Strength is a weakness when it comes to programming languages. The stronger and more expressive a programming language is, the more complex its code becomes. … Complexity is the enemy of security. Complex programs are more difficult to reason about and harder to identify edge cases for. … The less the language lets you do, the easier it is to analyze and prove properties ... The fewer possible attack vectors you have to consider, the easier it is to develop a secure contract. A simpler programming model also allows things like formal verification and automatic test generation." (Qureshi, 2017)

Hai Zhuge observes that constrained expressiveness results in a language that is faster to compute, and is more easily engaged by non-programmers:

> "The less information a semantic path contains, the easier people understand and remember. This indicates the simplest emerging principle: the shortest path with least types of semantic links takes priority to emerge as the relation between two nodes. This can be explained by Shannon and Weiner's theory of information entropy: the lower entropy a path has, the less semantic link types it contains, therefore it can be more easily understood." (Zhuge 2010, p. 202)

The expressive complexity of a programming language increases the computational work required to process data represented in that language. (Bush & Meyer, 2002). For any system intended for ubiquitous deployment, the least power orientation is relevant to the impact of data centre activity upon regional energy supply/demand issues (Ansar et al., 2019), and to decentralized computing capabilities when dependent upon batteries. This literal view of the ‘least power norm’ has largely been overlooked in informatics, even though it is a routine consideration in most other fields. But a small number of studies has emerged in recent years to compare the energy use implications of alternative programming languages and styles (Abdulsalam et al., 2015) (Lima et al., 2016) (Pereira et al., 2017) (Couto et al., 2017). Project management operations research is oriented to the least possible materials, labour, and energy required to design, implement, attain and maintain desired qualitative and quantitative outcomes.

2.5.2.6 Tabular Declarative Style

The *tabular declarative* programming style involves the unbundling of data and logic into discrete control tables, while building procedure into the platform. (Cunneyworth, 1994) This is distinguished from the much more popular *procedural imperative* style of programming in which one writes out all the logical procedures and data in the sequential way that humans tend to think about them. The tabular declarative style originated in the 1950s and 60s when transmission and computing hardware were rudimentary by comparison with today’s capabilities. Data travelled linearly via wires (telegraph) and radio waves (telex).[9] Given those earlier limitations, the programming style had to optimize the data structures to suit the machines, over programmer comprehensibility. Eventually, with increasingly faster hardware and networks, the procedural imperative style enabled programmer-friendly code. Easier programming led to more programming. However, larger programs and less compact data formats were needed to accomplish the same functions. Since computer hardware and network performance was advancing geometrically (‘Moore’s Law’) (Moore, 1965) (Moore & Courtland, 2015), software bloat was not usually presenting much of a problem.

In the past decade though, the proliferation of devices and networking has made high-performance tabular declarative programming interesting again. Fast and efficient parallel processing is required for high-volume, on-demand, event-oriented computation. In both centralized and decentralized deployments the tabular declarative style scales without adding complexity. Input and output data are represented via simple ordered lists (tuples). This can be employed for very simple data matching and sifting processes, which are orders of magnitude faster to process than trying to perform the same functions with conditional commands that would need to be parsed and processed in a step-by-step sequence (Cunneyworth, 1994) (Coenen, 1999) (Garcia et al., 2000) (Dean & Ghemawat, 2008b).

9 In the late 1800s "electronic" (electrical informatic) transmission was done on telegraph wire networks. The telex (teleprinter exchange) radio network was predominant from the 1940s to the 1970s. And in the early 1960s "packet switching" was designed (breaking messages into segments each with a header and a payload for transmission over a distributed variable network, for reassembly at the destination) and came to be implemented in the 1970s.

John Lloyd explains that separating logic from control enables the declarative style:

> "One of the main ideas of logic programming, which is due to Kowalski, is that an algorithm consists of two disjoint components, the logic and the control. The logic is the statement of what the problem is that has to be solved. The control is the statement of how it is to be solved. Generally speaking, a logic programming system should provide ways for the programmer to specify each of these components. However, separating these two components brings a number of benefits, not least of which is the possibility of the programmer only having to specify the logic component of an algorithm and leaving the control to be exercised solely by the logic programming system itself. In other words, an ideal of logic programming is purely declarative programming." (Lloyd, 1987. p 2)

Table 3 summarizes the implications of these different programming styles in relation to portability, security, scalability, efficiency and user knowledge.

Table 3: Differences Between Procedural Imperative and Tabular Declarative Programming Styles

	A Procedural Imperative Programming Style	**A Tabular Declarative Programming Style**
Portability	Code in particular procedural languages (such as C++, Java, Python, R) needs to be transcribed and recompiled to run on multiple platforms.	Cross-platform JavaScript Object Notation (JSON) can be embedded in many languages (such as C++, Java, Python, R), with just a fast, simple parsing library.
Security	General purpose programming languages are employed due to their rich expressiveness. As these are Turing-complete, the attack surface is large.	Deliberately low expressiveness is a mandatory design requirement such that this style MUST NOT be Turing-complete, thus facilitating intrinsic security.
Scalability	Algorithms are expressed as sequences of imperative conditions and procedural instructions to implement nested IF-THEN-ELSE inferential logic statements. These scripts and objects require compute-intensive processing with rigid inter-dependencies. This can scale on centralized systems, but is not efficiently scalable for ubiquitous distributed deployment.	Algorithms are expressed as declarative tuples (ordered lists, control-tables, truth-tables, logic gates, data-frames, grids) to implement normalized GIVEN-WHEN-THEN descriptive logic for simple data sifting and transformation. Tabular condition-assertion relations, separated from data, enable fast data-intensive parallel processing. This is scalable for ubiquitous distributed deployment.
Efficiency	Upon arrival of new input data, a procedural imperative sequence with an unknown number of ' applicable ' instances would run a metadata filter, then run conditions on each instance to establish which are 'true' or 'false'. This is inefficient for large scale, high volume, on-demand, event-oriented computation.	Upon arrival of new input data, a tabular declarative operation is executed only after the 'in effect' (context) sifting and 'applicable' (particulars) data sifting selects one or more 'hit' instances. All 'miss' instances are ignored. This is highly efficient for large scale, high volume, on-demand, event-oriented computation.
Knowledge	The user needs prior knowledge of what algorithms are in effect, applicable, and invoked.	The user needs no prior knowledge of what algorithms are in effect, applicable, and invoked.

Chapter 3: A Reflective Review of Literature on the Nature of a 'Rule'

3.1 What is a Rule?

A rule is any directional relation communicated among two or more people to associate what 'is' and what 'ought' to be. David Hume famously argued three centuries ago that "vice and virtue are not discoverable merely by reason" (Hume, 1738, p. 470),[10] or as this has been more widely expressed, one cannot derive 'ought' from 'is' (Spielthenner, 2017). But I suggest that anyone can assert a direction from 'is' to 'ought', which we portray here with the formal expression: 'IS + RULE $\Longrightarrow$ OUGHT'. In practical, logical, ethical and aesthetic matters, rules express obligation, permission or encouragement through the commonly capitalized terms MUST, MAY and SHOULD, or their various negatives and synonyms.

In the mid-1700s Samuel Johnson's Dictionary of the English Language referred to a "rule" as a precept by which thoughts or actions are directed through authority, influence or power. (Johnson, 1755) Johnson's concise definition emphasizes that a rule embodies a normative standard for behaviour, and yet, reliance upon authority, influence or power implicitly acknowledges that each individual subject to the rule retains free agency, that's to say, autonomous discretion, about whether or not, or to what degree, they will act in accordance with it. (Conrad, 1988) Otherwise the authority, influence or power wouldn't be needed, since conforming with the rule would be akin to unconscious breathing.

Formally standardized expression of rules in official documents of industry, commerce and governance arose contemporaneously with modern philosophical foundations of normative theory in the 1950s and 1960s (Wittgenstein, 1953) (Kripke, 1982) (Bloor, 1997) (Baker & Hacker, 2009). Our design research is guided in particular by Ludwig von Wittgenstein's "Philosophical Investigations" (Wittgenstein, 1953),[11] as elaborated in Georg Henrik von Wright's "Deontic Logic" (Von Wright, 1951), Jerzy (Georges) Kalinowski's "Theorie des propositions normatives" (Kalinowski, 1953) and Gertrude

10 At greater length Hume wrote: "In every system of morality, which I have hitherto met with, I have always remarked, that the author proceeds for some time in the ordinary way of reasoning, ...when all of a sudden I am surprised to find, that instead of the usual copulations of propositions, is, and is not, I meet with ...an ought, or an ought not. This change is imperceptible; but ...this ought, or ought not, expresses some new relation or affirmation, 'tis necessary that it should be observed and explained; ...for what seems altogether inconceivable, how this new relation can be a deduction from others, which are entirely different from it." (Hume, 1738, p. 463)

11 Norman Malcolm opens his article "Wittgenstein on Language and Rules" with exasperation: "A paradoxical situation exists in the study of Wittgenstein. There is a sharp disagreement in the interpretation of his thinking about the concept of following a rule." (Malcolm, 1989) Norman explains that some prefer Wittgenstein's later notion of a rule as requiring agreement amongst people about a guide to action; whereas others prefer Wittgenstein's earlier idea that a rule requires only repeated instantiation through action, even if performed by a single person. Norman provides his own position, but the essential point is that that there is no rule about what a rule is. Wittgenstein's incompatible 'earlier' and 'later' views each developed a following, so perhaps he can be blamed for the discord. (Wittgenstein, 1991) (Coeckelbergh & Funk, 2018) (Roermund, 2013) (Fielding, 2013) (Miller & Wright, 2002)

Anscombe's "Intention" (Anscombe, 1957).[12] These philosophers distinguished the *imperative*, *declarative* and *empirical* modes of expression of rules that are 'in effect' for a general context, that are 'applicable to a class of circumstance, and that are 'invoked' by particular events:

- A rule itself is an *imperative* statement of obligation, option or expectation among people;
- Documentation about a rule (a 'normative proposition') is a *declarative* statement of fact;
- Applicability of a rule to particular event is an *empirical* statement of deduction.

Table 4 builds upon distinctions by Jaag Hage (Hage, 1999) for communicating normative data.

Table 4: Six types of normative data which may be communicated.

NORMATIVE DATA *MUST, MAY and SHOULD*	**ASSERTION** *Empirical Statements*	**PROPOSITION** *Declarative Statements*	**PREROGATIVE** *Imperative Statements*
GENERAL CONTEXT	Assert from evidence that this rule set is 'in effect' for this jurisdiction and time, and is 'applicable' to such a circumstance.	Describe a system of rules.	Acknowledge a system of rules.
PARTICULAR EVENT	Assert from evidence that this rule is 'invoked' by these facts.	Describe a rule.	Acknowledge a rule.

Ideally within a "deontic network" (Quirico, 2009) anyone may readily discover, fetch, scrutinize, prioritize propositions about the rules of ethics that are 'in effect', 'applicable' and 'invoked' with the particulars of any "deontic circumstance" (Castañeda, 1989). The same would hold for goal-oriented rules within a normative network. But legal philosopher Carlos Alchourrón has cautioned that declarative propositions (statements) about rules of obligation, option or expectation among people cannot comfortably be assumed to be complete, accurate and consistent in their communication or application (Alchourrón, 1969). Von Wright credited Carlos Alchourrón with correctly accommodating gaps and inconsistencies into Kalinowski's structure for norm-propositions : "The

12 In his seminal 1951 paper entitled "Deontic Logic" Georg Henrik von Wright introduced a new formal mode of philosophical logic concerned with the deontic mode of logic (concerned with ethical principles and guidelines), placing it on par with alethic (**αλήθεια** means "truth") and epistemic (**ἐπιστήμη** means "knowledge") modes of logic (Von Wright, 1951). Independently and very shortly afterwards, Kalinowski published an article "Theorie des propositions normatives" in which he distinguished normative functions, that is to say norms themselves, from normative propositions, or descriptions of norms: "nous distinguerons entre fonctions logiques normatives et propositions logiques normatives" (Kalinowski, 1953). From these beginnings, the philosophical domain of ***normative relations*** and ***deontic virtues*** has embodied the original framing of both von Wright and Kalinowski, as evidenced by the name of the bi-annual "International Conference on Deontic Logic and Normative Systems" (DEON, 2019). The general framework shared across theoretical and applied research in this domain has come to be known as "standard deontic logic" (SDL), and this has continued to expand through inter-disciplinary collaboration involving theorists and practitioners in law, ethics, informatics and mechatronics. Kalinowski emphasized in 1953 that *norms* are commands communicated in an imperative, prescriptive context, while *propositions about norms* are reports given in a declarative, descriptive context. The human psychological tendency to conflate the descriptive *signifier* and the prescriptive *signified* is discussed by von Wright in relation to permissions and obligations: "Deontic sentences exhibit a characteristic ambiguity. One and the same form of words may be used for giving a norm and for stating that a norm to such and such effect has been given (exists). In the former case the sentence is used prescriptively, in the second descriptively. … Used prescriptively it imposes [norms]; used descriptively it gives information about existing [norms]. In the prescriptive use the sentence does not say anything which is true or false. In the descriptive use it does." (Von Wright, 1999, p. 31)

merit to have clearly realized this belongs to our deceased colleague and friend Carlos Alchourron... the first to devise a Logic of Norm-Propositions in contradistinction to a Logic of Norms.” (Von Wright, 1999, p. 20) Alchourrón commented that philosophers have paid insufficient attention to normative propositional logic, and thirty years later von Wright repeated: “I am afraid that later logicians have not always recognized this double task of the logical study of normative discourse. … Both Kalinowski and myself, I think it is true to say that, our continued work has with time distanced itself from that I would call 'mainstream' developments” (Von Wright, 1999, pp. 20, 21). This led von Wright to concede that rules as stated and applied typically fall short of the rigour that is achievable in formal deontic and normative logic:

> "Thus a classic deontic logic [of] formulas, pictures a gapless and contradiction-free system of norms. A factual normative order may have these properties, and it may be thought desirable that it should have them. But ...[e]xperience seems to testify that mutually contradictory norms may co-exist within one and the same legal order, and also that there are a good many 'gaps' in any such order ...This being so, classical deontic logic, descriptively interpreted, cannot claim to be the (correct) logic of norm-propositions. (Von Wright, 1999, p. 20)

Rules in a context of reflexive agency are expressed as assertions in which the modal auxiliary verbs *shall (must), should (ought to)* or may *(can)* are employed to connect foreseeable circumstances with various boundaries and options that are within the capabilities of those who are deemed subject to the rule. The terminology is sometimes ambiguous, however semantic precision is required to communicate whether an assertion expresses a requirement, recommendation, permission or capability. We borrow the “preferred verbal forms” published in the ISO/IEC “Principles and Rules for the Structure and Drafting of International Standards” (ISO/IEC, 2018, sec. 7). In Table 5, preferred terms in English are accompanied by several equivalent phrases or expressions for use in some cases when the preferred term seems less fluent in linguistic style.

It is convenient to characterize rules with “*is-ought*” terminology, however this does not require taking any stance whatsoever in the 250-year-long philosophical debate begun by David Hume about whether normative implications can ever be deduced solely from positive facts, or whether positive facts in any real context inevitably inherent normative attributes (Hume, 1738) (Spielthenner, 2017) (Roth, 2011) (Elqayam & Evans, 2011) (Kupperman, 2005).

In theoretical literature, as well as among professionals, the phrases ‘deontic logic’, ‘normative logic’ and ‘modal logic’ are used inconsistently. The terms are employed here in the manner clarified by Jerzy (Georges) Kalinowski. He distinguished the ***normative directionality*** of obligation, permission and encouragement in relation to goals among agents and their actions, versus their ***deontic virtues*** of obligation, permission and encouragement in relation to ethics among agents and their actions

(Kalinowski, 1953) (Trypuz & Kulicki, 2015, p. 391). The Latin term ***norma*** means ‘rule’ to express obligation, permission or encouragement, using the terms MUST, MAY and SHOULD, or their various synonyms and negatives. The Greek term ***δεον***, which means ‘duty’, adds qualities of moral goodness. These two aspects, *normative direction* and *deontic virtue*, are implied when referring to these terms as *modal* verbs, from the Latin word ***modus*** for ‘modality’ or ‘disposition’.

Rules are integral to every enduring human relationship, horizontal or hierarchical, at the societal, organizational, community or interpersonal level. Each rule is grounded in relative social and institutional agency with the weight of authority, agreement or preference. A rule can only be deemed to exist when communicated between ‘rule-maker agents’ and ‘rule-taker agents’, directly or mediated by their machines. When using such terms here, no assumption is implied about power relationships. The social ‘prerogative’ to establish rules may involve relations of *subsidiarity* based upon the discretion of the relatively disaggregated micro decision-makers, or *paramountcy* based upon the discretion of the relatively more comprehensive or macro decision-makers. In various contexts, human agency is oriented in hierarchical-tree, hub-and-spoke and peer-to-peer structures. Rule systems are complex and dynamic, like the human relationships they emerge from.

Any given rule is *'in effect'* only within particular dates/times and jurisdictions, and is *‘applicable’* only to certain categories of state condition. And it is only when particular events or circumstances are anticipated or occur which correspond to the input conditions of that rule, that one or more output assertions of the rule are *'invoked'*.

Some rules are concise, simple and clear, whereas others are expressed vaguely, leaving uncertainty about what exactly is required, allowed or expected. Some premises or requirements of a rule have to be left implicit or indirectly specified, otherwise they would get mired in excessive text. Multiple rules may be synergistic, or they may contradict or over-ride one another. There may be uncertainty regarding the gravity of non-conformance, and even about whether conformance is considered to be essential. (Verhulst et al., 2013) Therefore each person who is subject to a rule ultimately has discretion about whether or not, and to what degree, to act in accordance with it. (Bartlett, 2018)(Conrad, 1986) Without this premise of subject discretion, the authority, agreement or preference would be superfluous, since unthinking conformance with rules would be akin to machinery. Conformance and enforcement is usually tempered by the avoidance of ‘too much’ rigidity, and of ‘too much’ discretion. For all of the above reasons, rules are generally subject to interpretation, opposition and some degree of non-conformance. In what Jamie Bartlett called a *‘machinocracy’*, algorithmic rule-makers over-power rule-takers (Bartlett, 2018), so that there are no rules, only commands.

Table 5: Normative Verbs for Expressions of Requirement, Recommendation, Permission & Capability: "Principles and Rules for the Structure and Drafting of International Standards" (ISO/IEC, 2018)

A Requirement

- **shall**
  - is required to
  - it is required that
  - has to
  - only ... is permitted
  - it is necessary
- **shall not**
  - is not allowed [permitted] [acceptable] [permissible]
  - is required to be not
  - is required that ... be not
  - is not to be
  - do not

A Recommendation

- **should**
  - Equivalent phrases or expressions
  - it is recommended that
  - ought to
- **should not**
  - it is not recommended that
  - ought not to

A Permission

- **may**
  - is permitted
  - is allowed
  - is permissible
- **may not**
  - it is not required that
  - no ... is required

Possibility and capability

- **can**
  - be able to
  - there is a possibility of
  - it is possible to
- **cannot**
  - be unable to
  - there is no possibility of
  - it is not possible to

External constraint

- **must**
  - Do not use "must" as an alternative for "shall". This avoids confusion between the requirements of a document and external constraints.

Rules can usually be identified by the use of various standardized terms. The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) specify a set of normative verbs in"Principles and Rules for the Structure and Drafting of International Standards" (Table 5) (ISO/IEC, 2018). In the 1990s the Internet Engineering Task Force (IETF) similarly formalized the capitalized terms MUST, MAY and SHOULD as key words to indicate requirement levels (Bradner, 1997). Across all subject domains, not only in informatics, the IETF document has become the most widely referenced standard for these terms in rule expression. In the present work we employ these capitalized terms in conformance with the ISO/IEC guidance for any 'external constraint', which maps to the IETF usage.

Groups of people compose sets of norm-propositions as various kinds of rules: laws of jurisdictions, contracts between parties, standards across industries, protocols and customs of communities. Contradictory rules may coexist, and a particular circumstance might have more than one 'correct' interpretation, but efficient and effective communication of norm-propositions helps towards recognizing and resolving such dissonance. Every system of rules evolves through distinctive social and historical circumstances, involving myriad factors that interact through direct and indirect feedback loops.

Although systems of rules (norms) are complex, eclectic, and hard to comprehend, a system for the transmission of information about rules (norm-propositions) can at least be simple and reliable. Our purpose here is to design and make available a common efficient way to communicate norm-propositions from *rule-maker agents* to *rule-taker agents*.

We distinguish *normative logic of* obligation, permission and encouragement from both *deductive logic* of epistemology, and *procedural logic* of method. The three can be compared by adapting a simple declarative input-output relation:

- *Deductive I/O Logic* asserts a reasoned conclusion from a premised truth.

  LET A_l and A_2 and ... and A_m be true
  THEN B_l and B_2 and ... and B_n are true

  Example:

  LET:

  Al = "Data was collected in an ethical manner";

  A2 = "Data was protected from misuse";

  THEN:

  Bl = "Data privacy policies would have been respected"

  B2 = "Data security measures would have been in place".

  The causal term ‘THEN’ expresses a deducible conclusion, meaning *therefore*, like the French term *alors*, and the Spanish term *entonces*. This can be summarized with truth tables.

- *Procedural I/O Logic* asserts a programmable instruction from a contingent empirical state.

  IF A_l and A_2 and ... and A_m present
  THEN B_l and B_2 and ... and B_n execute

  Example:

  IF

  Al = "Data is collected from users";

  A2 = "Data is used for decision making";

  THEN

  Bl = "Run data accuracy filters";

  B2 = "Run data validation checks for completeness";

  B3 = "Run data storage security routines";

  B4 = "Run data integrity monitor for changes and anomalies"

  The temporal term 'THEN' expresses an actionable sequence, meaning *next,* like the French term *ensuite*, and the Spanish term *luego*. This can be summarized with decision tables.

- *Normative I/O Logic* asserts obligation, permission or encouragement from a reported empirical state.

  WHEN A_l and A_2 and ... and A_m is
  THEN B_l and B_2 and ... and B_n ought to be

  Example:

  WHEN

  Al = *"Data is collected from users";*

  A2 = *"Data is used for decision making"*

  THEN

  Bl = "Data accuracy filters MUST be run";

  B2 = "Data validation checks for completeness MUST be run";

  B3 = "Data storage security routines MUST be run";

  B4 = "Data integrity MUST be monitored for changes and anomalies"

The directional term 'THEN' expresses a social motive, meaning *so,* like the French term *donc*, and the Spanish term *pues*. This can be summarized with tables designed for data with direction.

These three are commonly conflated in the published literature, since the techniques employed to resolve and to communicate all such logic problems are the same. Therefore, our historical review in the next section draws upon contributions from all three sub-domains of the field without differentiation, however the rest of this work on the design specification for a processing pipeline for normative data with direction relies upon terminology and methods that are clearly distinguishable from deductive and procedural types.

3.2 What is Agency?

3.2.1 Direction from ‘Is’ to ‘Ought’

In the present work the term ‘agency’ refer to use of one’s attitudinal, intellectual and tangible faculty of action to pursue a specified result. It has a ‘direction’ in sense express by Alexander Riegler in the inaugural issue of the journal *Constructivist Foundations*: “reality is brought forth by the subject”. (Riegler, 2005) Mustafa Emirbayer and Ann Mische emphasize that this is reality brought forth from the person’s inner capacity for iterative patterns of thought, imagination of actions, and judgment of consequences (Emirbayer & Mische, 1998). Margaret Archer expands upon the imagination part as involving three phases of the inner conversation (reflexivity) which the agent undertakes:

1. Defining and aligning a set of concerns;
2. Developing concrete projects (courses of action), and on that basis
3. Establishing practices (Archer, 2007, p. 8).

Paul Gaddis elaborates the concrete project part, defining the organized agent as a “project manager” seeking to achieve a planned outcome “on time, within budget, and in conformance with predetermined performance specifications” (Gaddis, 1959).

Not every agent is as well-organized, but everyone exercising agency is engaged in transforming what ‘is’ in the direction towards what they consider ‘ought’ to be. Whenever this becomes a persistent joint social endeavour among a cluster of people who organize themselves, or are organized, to operate as a social undertaking or entity, then they develop various practical, logical, ethical and aesthetic norms. The consequent obligations, permissions and encouragements that they come to experience internally and communicate externally in relation to those norms, rooted in agency, are ‘rules’.

3.2.2 Source, Subjectivity and Strength

Presumably one ought to abide by legitimate rules. And yet, not conforming with an applicable rule in a particular scenario might not imply violating it.

Economist Robert Barro points out in a paper on the “theory of rules versus discretion” (in reference to monetary policy, but with general relevance) that the choice about following or not following some rule is usually a choice amongst types of rules, rather than between rules or no rules (Barro, 1986, p. 32). This is explained with clarity by military ethicist Richard Gabriel. His is referring to the special circumstances of war, but he also offers generally valuable insight:

> "It is certainly true that some soldiers may choose to observe different obligations of the code even in similar circumstances. But this is the very nature of making ethical choices. The reason why a soldier chooses one obligation over another is because the soldier judges that one obligation in a given set of circumstances ought to take precedence over another precisely because he or she deems it to be more valuable. Nonetheless, all postulates of the code are *prima facie* binding. It is the circumstances in which the precepts have to be applied that force the soldier to render a judgment that one precept has greater worth than the other and must be observed first. But as precepts *per se*, they are all equally ethically imperative insofar as they require the obligations to be observed *if the circumstances permit*. They are also equally imperative in the sense that they can be raised by the principle of universality so that if all soldiers carried out all the precepts *per se*, we would judge their actions as ethical. *The fact that one soldier may value one obligation over another in a different set of circumstances than another soldier does not negate the value of the code in stating what obligations ought to be observed in the first place."* (Gabriel, 2007, p. 170)

Unbridled discretion risks impulsive or opportunistic behaviour; yet too much compliance pressure risks inculcating overly rigid codes of behaviour (Espedal, 2007). A general purpose system for transmitting rules needs to be capable of rule triage. Not only do rules need to be classified, they must come packaged in a sufficiently meaningful 'envelope' of metadata as to characterize the particular rule relative to other rules.

To enable discretion in weighing a rule relative to any other rule, the metadata envelope delivering a given rule could include the three context variables suggested by academic linguistics specialists An Verhulst, Ilse Depraetere and Liesbet Heyvaert (Verhulst et al., 2013):

- *Source*: *de jure* authority and/or *de facto* origins of a regulation, condition or target circumstance;
- *Subjectivity*: the involvement or commitment of beneficiaries towards ensuring fulfillment;
- *Strength*: the gravity or the impossibility of non-compliance.

These three variables integrated with the options listed in *"Who Rules"* provide a practical basis for nuanced prioritization of rules. Three source-subjectivity-strength columns of a table can be populated with whole numbers between 0 and 100 (where null defaults to 0, meaning 'experimental'). This level of precision would "pave the way for a detailed data analysis that manages to pin down the related yet distinctive shades of meaning of ... root necessity." (Verhulst et al., 2013, p. 221). The allocation of the 101 (0 to 100) rankings could potentially be governed somewhat as radio frequencies are allocated under the International Telecommunication Union, based on a set of criteria and decision processes (ITU, 2016, sec. Article 1.16, definition: allocation of a frequency band). The governance body would need to have formal government participation, as the ITU does, unlike the peer-oriented IETF, simply because the apportioned ranking includes jurisdictional statutory prerogative of governments.

Verhult et.al. developed their source-subjectivity-strength framework to the analysis of English language semantics. These three factors seem to be generally useful in a multi-factor ranking rules in a global multicultural and multilingual setting for general purpose rules transmission and use.

This approach might even enable the inclusion of sectors of society that function entirely through customary relationships (Cao, 1999, p. 162). This is because in addition to rules expressed in source code, an Internet of Rules could contextually fetch and deliver to users rule documents expressed as principles, decisions, stories and allegories.

A convenient view of the diverse contextual settings in which rules occur is provided by David Weissman in his book *The Cage: Must, Should, and Ought From Is*:

> "Context has eleven variables: i. systems in which one participates; ii. personal health or illness, and the health of one's core or other systems; iii. economic organization and productivity; iv. civil peace and stability; v. physical topography and climate; vi. culture; vii. technology; viii. conventional laws; ix. logical and natural laws; x. personal and social history; and xi. conflict." (Weissman, 2006, p. 241)

Context is crucial, since a behavioural response to a particular rule may be conditioned by:

- Actual or apparent contradiction amongst equally valid and reasonable rules (antinomy) (Thornhill, 2012) (Eastman & Bailey, 1998);
- Circumstantial inability (Kurthy et al., 2017);
- Overriding obligations (Litman, 2003) (Gabriel, 2007);
- Uncertainty (Lei & Coulton, 2011) (Carlin, 2009);
- Unenforceability (Avelhan & Zylbersztajn, 2018) (Lindberg, 2018) (Zhou & Poppo, 2010);
- Systemic dysfunction (Adams & Balfour, 2015) (Alvesson & Spicer, 2012) (Dillard & Ruchala, 2005);
- Civil disobedience (Thoreau, 2008) (Brownlee, 2012);
- Uncivil obedience and 'work-to-rule' (Bulman-Pozen & Poz, 2015) (Markovits, 2016) (Simon, 2009, pp. 90-91);
- Counter-intuitive benefits in certain scenarios when authorities choose to condone some amount of rule-breaking (Eigen et al., 2015) (Murray, 2015) (Peñalver & Katyal, 2010) (Robbins et al., 2006) (Goodin, 2005).

3.3 What is an Algorithm?

3.3.1 A Precise and Composable Extension of Human Agency

Samuel Johnson described an "algorithm" as being a set of operations in the science of numbers. (Johnson, 1755) This is consistent with the original word "algorism" (with an 's') which, since the 12th century referred to the methods for using the nine Hindu-Arabic numerals 1, 2, 3, 4, 5, 6, 7, 8, 9 and the cypher 0, as first described by the 9th-century scholar Abi Jacfar Muhammad ibn Miisa al-Khwiirizmi [al-Khwsrizmi]. (Crossley & Henry, 1990) It is generally agreed that the words *algorism* and *algorithm* arose from the Latin transcription of al-Khwiirizmi's Persian name.

The modern term 'algorithm' (with 'th') has three distinct meanings in the literature : a practical intuitive definition, and two precise mathematical statements. In its practical meaning, an algorithm is a set of operations invoked by a condition, to carry out a procedure to solve a general, well-specified problem, that terminates once the procedure has run. (Skiena, 2008, p. 3)

The two formal schools of thought on what an algorithm is are led by Yuri Gurevich and by Yiannis

Moschovakis (Vardi, 2012). In his paper "What is an Algorithm?" Gurevich defines it as any instruction for storing data in a particular state, and then based on input data x, generating a resulting output f(x). He refers to this as an "abstract state machine" (Gurevich, 2014). Moschavakis, in his paper also entitled "What is an Algorithm?" defines it as any computable expression on a Turing machine designed to accept input data x, and to return output f(x) as further input, then terminate once a specified condition is reached. He refers to this as a "recursor" (Moschovakis, 2001). Moshe Vardi, Editor-in-Chief of *Communications of the ACM*, accepts both perspectives in his editorial aptly headlined : “What is an Algorithm?”

> “So is an algorithm an abstract state machine or a recursor? Mathematically, one can show that recursors can model abstract state machines and abstract state machines can model recursors, but which definition is primary? … An algorithm is both an abstract state machine and a recursor, and neither viewed by itself fully describes what an algorithm is. This algorithmic duality seems to be a fundamental principle of computer science.” (Vardi, 2012, p. 5)

Bert Van Roermund explains the difference between an algorithm and a rule by distinguishing the operational rows of a computational table, versus the rationale of the whole table. The entire context and purpose of the table constitutes a rule, whereas each operational row in the table is a discrete algorithm to be computed under certain conditions to invoke the rule. He explains that an agent running any of the algorithms “will only be following the rule if they see ‘the point’ of the row. But neither the rows nor the columns supply this point”. (Roermund, 2013) .

He further illustrates this with the simple example of an algorithm that produces the number sequence 1, 2, 3, 4. Some underlying rule is needed to determine what algorithm may be coded to generate the number, or next few numbers. The problem is that there is not sufficient information to know what that rule might be. An obvious answer "5, 6, 7, 8, 9, 10" will only be correct if it can be verified that the applicable rule is to continue along the set of natural numbers N*. When that is true, a suitable algorithm expressed in natural language could be declared as:

(a) Increase the previous number by 1, repeat 9 times, and then stop.

But as Roemund explains, some other rule might be ‘in effect’ and ‘applicable’:

(b) Repeat the previous 4 numbers for 3 cycles, and then stop.
(c) Generate a random combination or permutation of the initial 4 numbers, and repeat 8 times, and then stop.
(d) Increase the previous number by 1, and repeat 10 times but suppress all numbers divisible by 5, and then stop.

It is worth pointing out that there is no authoritative global publication of theoretical mathematics that provides the definitive statement on the natural numbers rule. However a practical statement is provided by industry in the ISO 80000-2:2009 standard on quantities and units, shown in Table 6, specifically Item No. 2-6.1 the set of natural numbers, denoted N to include 0, and N* to exclude 0

(ISO, 2009). Ironically, no industry manager would ever check this ISO standard (priced at 158 Swiss francs!) when using the natural numbers rule. There are plenty of good sources on arithmetic for a manager to rely upon should there be a need to actually check the definition. (Does it include “0”?) And yet it is useful to cite this standard when programming an official algorithm precisely because it ought to be backed by consensus across a community of knowledgeable stakeholders. Any informed thoughtful person can quickly scan some ‘good’ sources and find the more common natural numbers rule that uses the symbol ℕ instead of N*. To have a computer programmed for which of these symbols to use would require some context. Instead, a simple look-up to a table of relevant standards is enormously more efficient (...or it would be if ISO were to create for itself an Internet-age revenue and dissemination model).

Table 6: Excerpt from ISO. (2009). ISO 80000-2:2009. Quantities and units. Part 2: Mathematical signs and symbols to be used in the natural sciences and technology

ISO 80000-2:2009(E)

6 Standard number sets and intervals

Item No.	Sign, symbol, expression	Meaning, verbal equivalent	Remarks and examples
2-6.1 (*11.4.9*)	**N**	the set of natural numbers, the set of positive integers and zero	$\mathbf{N} = \{0, 1, 2, 3, \ldots\}$ $\mathbf{N}^* = \{1, 2, 3, \ldots\}$ Other restrictions can be indicated in an obvious way, as shown below. $\mathbf{N}_{>5} = \{n \in \mathbf{N} \mid n > 5\}$ The symbols $\boldsymbol{N}$ and $\mathbb{N}$ are also used.
2-6.2 (*11.4.10*)	**Z**	the set of integers	$\mathbf{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ $\mathbf{Z}^* = \{n \in \mathbf{Z} \mid n \neq 0\}$ Other restrictions can be indicated in an obvious way, as shown below. $\mathbf{Z}_{\geqslant -3} = \{n \in \mathbf{Z} \mid n \geqslant -3\}$ The symbol $\mathbb{Z}$ is also used.
2-6.3 (*11.4.11*)	**Q**	the set of rational numbers	$\mathbf{Q}^* = \{r \in \mathbf{Q} \mid r \neq 0\}$ Other restrictions can be indicated in an obvious way, as shown below. $\mathbf{Q}_{<0} = \{r \in \mathbf{Q} \mid r < 0\}$ The symbols $\boldsymbol{Q}$ and $\mathbb{Q}$ are also used.
2-6.4	**R**	the set of real numbers	$\mathbf{R}^* = \{x \in \mathbf{R} \mid x \neq 0\}$

The well-established notion of “computer-human interaction” conflates user-interface design with agency. In a tug-of-war, the physical contact is between your hands and the rope, but the struggle is not between you and the rope; it’s between you and the person on the other end of the rope. Humans interact with other humans through radio signals, images, computer programs, and algorithms.

The computational algorithm should be understood as a precise and composable evolutionary extension of human agency. Whereas agency is the possession of attitudinal, intellectual and tangible faculty of action to pursue a specified result, an algorithm is a method invoked by a condition to obtain a specified result and then terminate. But whose agency is being extended: the algorithm user’s or the algorithm designers agency? Sven Nyholm refers to "collaborative agency" where people share agency with computerized machines (Nyholm, 2018), and Douglas Rushkoff speaks of relinquishing our agency to machines (Rushkoff, 2010, p. 14). However it seems a logical error to portray something designed and maintained by agents as itself possessing agency at all. Granted, a machine with advanced automation capabilities can convey an impression of agency. Yet it is under the control of its designers and programmers. A partial analogy is a movie character that may convey an impression of being, doing or feeling, yet the audience knows that it is really the script writer, director, actor and production team determine the character. Lisanne Bainbridge opened her influential 1982 paper "Ironies of Automation" with an observation about the people involved:

> The designer's view of the human operator may be that he is unreliable and inefficient, so should be eliminated from the system. There are two ironies of this attitude. One is that designer errors can be a major source of operating problems. ... The second irony is that the person who tries to eliminate the operator still leaves him to do the tasks which he cannot think how to automate. It is this approach which causes the problems to be discussed here, as it means that the operator can be left with an arbitrary collection of tasks and little thought may have been given to providing support for them. (Bainbridge, 1982, p. 129)

A responsible and autonomous manager may potentially determine that an algorithm is defeating the purpose of a rule that they are jointly supposed to implement. It could well be the manager who is mistaken. Or, possibly the algorithm has a software design or programming flaw, or it’s receiving faulty data (Ethiopia, 2019) (Indonesia, 2018) or it is compromised by an adversarial attack (Finlayson et al., 2019) (Knight, 2019) (Tencent, 2019). Commonly there may be more than one good way to accomplish the purpose of a rule, and the algorithm designer and manager may simply have different preferences or priorities.

One may consider this the other way around. Suppose that a manager determines that what’s needed to resolve a design limitation or functional flaw in essential equipment that lacks a sufficient degree of automation, is to write and implement a suitable algorithm to operationalize it. For example, Richard Stallman recounts the story of a group of system programmers at the Artificial Intelligence Lab at MIT in the 1980s who wanted to write and implement a simple algorithm to notify users, upon sending a print job, that the printer several floors away had a paper jam. But the manufacturer was reluctant to promote awareness of malfunctions such as paper jams, and refused. The owners and users of the printer were dis-empowered and over-ruled. The irony and fundamental ethical transgression of this obstruction started Stallman on the path to writing the free software principles, and creating a global movement to defend the agency of algorithm users (Stallman, 1991).

Agency is both constrained and enabled in what Robert Latham describes as a context of “social sovereignty” (Latham, 2000). Every agent lives in a society of reflexive agents and institutional structures.

Kees van Dongen and Peter-Paul van Maanen recommend the following practices to ensure that algorithm managers adequately support the operations managers who would rely on them (van Dongen & van Maanen, 2013):

Understanding
- Use simple modular algorithms to aid decisions
- Ensure operators know about each decision aid
- Make the reasoning of each decision aid easily available and understandable
- Reveal intermediate results in a comprehensible way

Accountability
- Ensure people feel accountable for performance
- Ensure people are held responsible for the quality of the outcome

Error Management
- Make potential sources of error, and actual errors, transparent
- Help operators to be aware of system biases, and their own biases
- Inform operators about conditions in which the decision aid performs well, and the conditions in which it does not

Such recommendations may seem as they can be taken for granted, but they come into focus when an operations manager’s judgment conflicts with the output of an automated or semi-automated system.

Much can be learned from the emerging standard ISO/SAE J3016 "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles". This technical framework describes six discrete and mutually exclusive levels of driving automation. "Central to this taxonomy are the respective roles of the (human) user and the driving automation system in relation to each other." (p.17) The six levels distinguish what parts of the dynamic driving task (DDT) are performed on a sustained basis by the human (device operator) and by the automation system (algorithm manager) (SAE, 2018).

- Driver performs part or all of the DDT
    - Level 0: No driving automation (driver leads unconditionally)
    - Level 1: Driver assistance (driver leads most tasks & conditions)
    - Level 2: Partial driving automation (allocated tasks & conditions)
- ADS (“System”) performs the entire DDT while engaged
    - Level 3: Conditional driving automation (fallback-ready driver)
    - Level 4: High driving automation (system leads with conditions)
    - Level 5: Full driving automation (system leads unconditionally)

We may adapt these levels to generic *unranked* allocations of prerogative, but retain the division based upon whether the operations manager’s or the algorithm manager’s agency predominates.

3.3.2 Algorithms, Agency and Prerogative

Earlier in the section "What is a Rule?" it was explained that the social 'prerogative' to establish rules can involve relations of *subsidiarity* based upon the discretion of the relatively disaggregated micro decision-makers, or *paramountcy* based upon the discretion of the relatively more comprehensive or macro decision-makers. We may then use the term "meta-rule" to refer to a rule which makes that allocation of prerogative explicit. So it would be a meta-rule that establishes whether an algorithm manager's agency is to be prioritized over an operations manager's agency, or vice versa.

A meta-rule could say that, to the extent the algorithm manager is not fulfilling a given requirement, the operations manager can, should or must over-ride the algorithm and take control. The other way around, the meta-rule might say that to the extent the operations manager is not fulfilling a given requirement, the algorithm manager can, should or must over-ride the operations manager and take control.

But which way is this to go? Whenever one manager's agency is explicitly prioritized over the other manager's agency, this *must* be accompanied with an explicit rule. And in either circumstance, errors and error-correction must be handled elegantly. Who can/should/must over-ride whom, under what circumstances, based on what criteria? Whatever the answer, how can this be ensured? There is no single 'best' model for the apportionment of agency between the algorithm manager and the on-site manager whose operations that algorithm is designed to improve. However there is need for precision in how the mutual prioritization of their agency shall work.

A common default meta-rule is that an on-site operations manager shall always hold priority agency and responsibility for operations within their mandate, including when employing automated and semi-automated aids. Table 7 "*Who Rules: The Algorithm Manager or the Operations Manager?*" provides a detailed framework of options for such structure. This table is currently in natural language text, but based upon design of DWDS in the present research, and following further peer review on its substance, it can be developed into a functional table that would operate as a human-accessible and machine-computable meta-rule for role-based prioritization of prerogative. This would enable certain rules to over-ride other rules.

All rules are assertions for reflexive contextual use by individuals and organizations; they are not linear mechanistic behaviour-controlling contraptions. A general-purpose Internet of Rules is being designed to empower people with knowledge about and access to all rules '*in effect*', '*applicable*' and 'invoked' for a circumstances, without any implication, suggestion or assumption that algorithm managers would be or have been authorized to pre-empt decisions by operations managers who may exercise their agency to use the algorithm, or not.

Table 7: Who Rules? The Algorithm Manager or the Operations Manager?

(This table re-frames, adapts and extends work by Ken Endo, 1994. p. 637, 641, 642)	
Operations Manager Agency is Prioritized	**Algorithm Manager Agency is Prioritized**
Empowering Statement	**Empowering Statement**
To the extent the algorithm is not fulfilling a given requirement, the operations manager: ○ *can* ○ *should* ○ *must* ...over-ride the algorithm and take control.	To the extent the operations manager is not fulfilling a given requirement, the algorithm manager: ○ *can* ○ *should* ○ *must* ...over-ride the operations manager and take control.
Constraining Statement	**Constraining Statement**
To the extent the operations manager is fulfilling a given requirement, the algorithm manager: ○ *cannot* ○ *should not* ○ *must not* ...over-ride the operations manager to take control.	To the extent the algorithm is fulfilling a given requirement, the operations manager: ○ *cannot* ○ *should not* ○ *must not* ...over-ride the algorithm to take control.
Delegating Statement	**Delegating Statement**
The operations manager *may* voluntarily delegate control to the algorithm manager: ○ *pro-actively* ○ *upon request*	The algorithm manager *may* voluntarily delegate control to the operations manager: ○ *pro-actively* ○ *upon request*
Criteria for Intervention or Delegation	
Potential criteria that could be applied to justify a reversal of agency in any of the above contexts: ○ *better attainment criterion* ○ *effectiveness criterion* ○ *efficiency criterion* ○ *informational or sequential criterion (in order to proceed)* ○ *pre-emption of conflict with higher priority rules criterion (necessity; cross-boundary; mandated)*	

Endo, K. (1994). The Principle of Subsidiarity: From Johannes Althusius to Jacques Delors. √ 이측정된상, *44*(6), 652–553. *https://eprints.lib.hokudai.ac.jp/dspace/bitstream/2115/15558/1/44(6)_p652-553.pdf*

A meta-rule provides the structure required to elegantly handle any multi-jurisdictional system of rules, as well as exceptions, errors, error-correction, omission and indemnification. There certainly are cases in which unconditional automation has been implemented unintentionally or inadvertently, and in which operations managers have ceded control only for semi-automated operation or tentative automation which lead to a weakening of their situational awareness. Even where functional control is clearly and deliberately ceded by operations managers to algorithm managers, the residual prerogative and agency of these two managers must be clearly established in policy and law.

- Operations Manager Agency Leads
  - Manual operation
  - Assisted operation
  - Semi-automated operation
- Algorithm Manager Agency Leads
  - Tentative automation
  - Conditional automation
  - Unconditional automation

Across all of these allocations of power, I suggest a generalized default policy similar to the editorial stance of the scientific journal Nature regarding the provision of access to algorithms in the articles they publish:

> "If the custom algorithm/software is central to the method ... it must be supplied by the authors in a usable form including one or more of the following.
>
> - Source code
> - Complete pseudocode
> - Full mathematical description of the algorithm
> - Compiled standalone software
>
> We strongly urge that full source code be provided. A compiled executable alone is not sufficient... Supplied source code or software must be accompanied by documentation sufficient for a typical user to compile, install and use the software. ... If appropriate, sample data known to work on the software should be provided along with the expected output. ... Any restrictions on the availability of software or code used to implement novel algorithms must be specified at the time of submission. ... We encourage authors to provide a license with the software or code. A narrative description of key algorithmic components should be provided in the main text. (Nature, 2014) (Evanko, 2014)

This is more succinctly and generally stated in the Free Software Definition, which I suggest to combine with my own four Principles of Accessible Algorithms:

Free Software Definition (Stallman, 1991)(Stallman, 1991)(Stallman, 1991)	**Principles for Accessible Algorithms**
Freedom 0: Freedom to run the program for any purpose.	Principle 1: Algorithms should be declarative.
Freedom 1: Freedom to study how the program works, and adapt it to one's needs.	Principle 2: Algorithms should embody patterns.
Freedom 2: Freedom to copy and redistribute the program	Principle 3: Algorithms should be published.
Freedom 3: Freedom to improve the program, and release any modified versions.	Principle 4: Algorithms should be simple.

3.4 Rule Transmission Systems

3.4.1 Signal and Noise in Rule Transmission Systems: Insights from Information Theory

In his 1948 paper Claude Shannon proposed a way to measure the amount of information in an electronic transmission. His general communication system model, shown in Figure 4, links information source and destination, with a source of noise (Shannon, 1948).

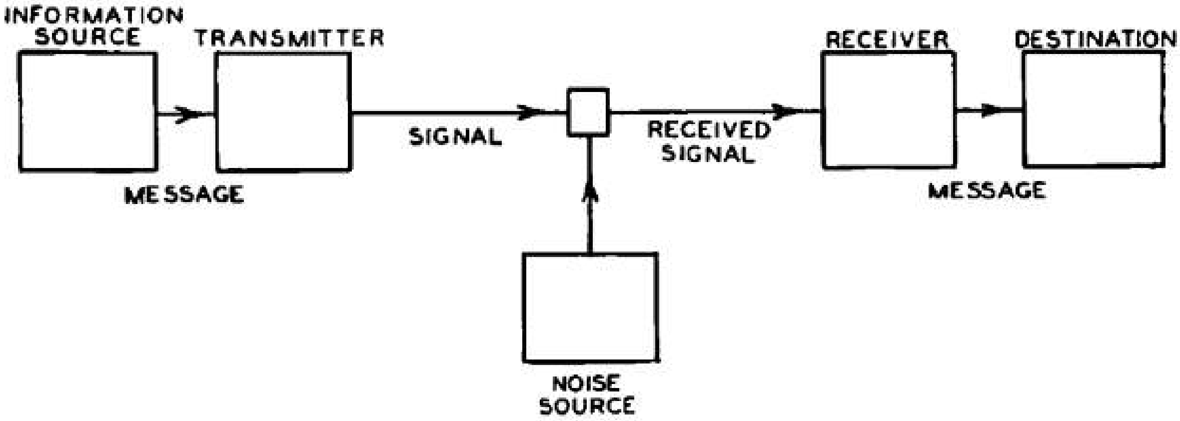


Figure 4: Schematic diagram of a general communication system (Shannon, 1948), reproduced directly from the original.

For our context, a rule is communicated as a normative proposition in a message, conveyed as a signal with some amount of noise between transmission and receipt. We may suppose the information source is anyone with social prerogative as a 'rule-maker agent', and the information destination is any 'rule-taker agent' deemed to be subject to these rules based on authority, agreement or preference. The rule itself is an experienced obligation, permission and encouragement in a social context among individuals and entities. So it is not the *imperative* rule *per se* which is the transmitted payload, but a *declarative* statement about a rule. The designer of a rule system enables the communication of normative propositions.

Normative propositions are communicated among two or more people to associate what 'is' and what 'ought' to be, but it is the quality of that communication, its '*signal-to-noise ratio'*, which underlies a message's performance in conveying any social sense of obligation, permission and encouragement. An ideal noiseless system of communication may be imagined, but all tangible signalling system implementations depend upon "ways of transmitting the information which are optimal in combating noise" (Shannon, 1948) (Shannon, 1998). Noise in such systems can arise at every incremental step from gaps, errors and inconsistencies in the descriptive expression, the transmission of the payload message, access to the transmission, and other factors. A rule system is better at reducing noise when it can aid in tracing gaps, errors and inconsistencies back to where they are introduced so that they can be averted, thus reducing dependence upon after-the-fact patches.

The challenge is that the information sources and destinations that frame the communication of what 'is' and what ‘ought’ to be, are clusters of contingent social entities, relations, jurisdictions and interpretations that are changing, dividing, merging, disappearing and being introduced, amidst complex hierarchical and distributed multi-objective optimization (Ehrgott, 2012, p. 448) and adaptive resilience to disturbance (Limnios et al., 2014, p. 106). Uncertainty is inherent in such complex systems.

In Shannon’s model the value ‘H’ represents the relative uncertainty of an observer about an outcome from amongst a set of possibilities, each of which have known probabilities of occurrence. If there’s only one possibility, that’s to say, if the observer can be entirely certain about that particular possibility being the outcome, then H=0. To express this abstract notion in terms of practical project management, Adrian McDonough explains:“the best decision-maker is one that will need the least new information to make the best decisions” (McDonough, 1963, p. 72). George Miller explains this in relation to the well-known concept of ‘variance’:

> “When we have a large variance, we are very ignorant about what is going to happen. If we are very ignorant, then when we make the observation it gives us a lot of information. On the other hand, if the variance is very small, we know in advance how our observation must come out, so we get little information from making the observation. (Miller, 1994, p. 343)

It’s tempting, therefore, to imagine a *‘know-it-all’* decision-maker as economist Kenneth Arrow did, adopting Shannon's notion of H=0 as a conceptual basis for modelling decisions with an assumption of “perfect information”. He wrote that: "if a channel of capacity H is installed, then the individual knows the state of the world (Arrow, 1984, p. 109)". Decision theory in much of the published academic literature in economics makes this working assumption. Or they assume some degree of H>0, meaning “imperfect information” that logically depends upon the H=0 referent.

However information theorist Edwin Jaynes presents an alternative to Shannon’s perspective. To begin with he emphasizes the inevitable uncertainty about the communication channel itself. If a channel of capacity H is installed, the observer will only gain access to knowledge about whatever this particular channel has been designed to convey. Jaynes explains that "Shannon's H measures the degree of ignorance of the communication engineer when he designs the technical equipment in the channel” (Jaynes, 1979, p. 38).

Jaynes also critiques Shannon’s premise that there can be a set of possibilities with known probabilities of occurrence. That clearly requires considerable prior knowledge. Jaynes explains:

> “There is no application of probability theory in which one can evade that all-important first step: assigning some initial numerical values of probabilities so that the calculation can get started. For even in the most elementary homework problems, such as ‘Find the probability of getting at least two heads in four tosses of a coin’, we have no basis for calculation until we make some initial judgment, usually that ‘heads’ shall have the probability 1/2 independently at each toss. But by what reasoning does one arrive at this initial assignment? (Jaynes, 1979, p. 16)

This is known as “Bernoulli's Principle of Insufficient Reason”, after Jacob Bernoulli's *Ars Conjectandi* (1713) (Shafer, 1996). John Maynard Keynes renamed this the “Principle of Indifference” (Keynes, 1921). Daniel Kahneman and Amos Tversky have demonstrated that people informally estimate the likelihood of a probability relation or frequency of occurrence of some event based on the ease with which its instances and frequency can be brought to mind. (Kahneman & Tversky, 1982) Even if the probabilities of Shannon’s set of possibilities are assumed informally, this constitutes an unexplained and unmeasurable prior amount of information in the model.

Jaynes thus argues that Shannon’s notion of ‘perfect knowledge’ H=0 is an impossibility. Not only does every eventuality in a set of possibilities enter consideration with unknown probabilities, but the conceptualization of the possibilities themselves is unexplained. In Jaynes’ view there is only one fixed point, which is 'perfect ignorance'. Analysis cannot begin with a *‘know-it-all’* omniscient decision-maker, but with a *‘know-nothing’* neophyte who asks questions and can learn. And this indeed provide a straightforward way to measure ‘bits” of information, as explained by Miller:

> “One bit of information is the amount of information that we need to make a decision between two equally likely alternatives. ... Two bits of information enable us to decide among four equally likely alternatives. Three bits of information enable us to decide among eight equally likely alternatives. Four bits of information decide among 16 alternatives, five among 32, and so on. That is to say, if there are 32 equally likely alternatives, we must make five successive binary decisions, worth one bit each, before we know which alternative is correct. So the general rule is simple: every time the number of alternatives is increased by a factor of two, one bit of information is added. (Miller, 1994, p. 344)

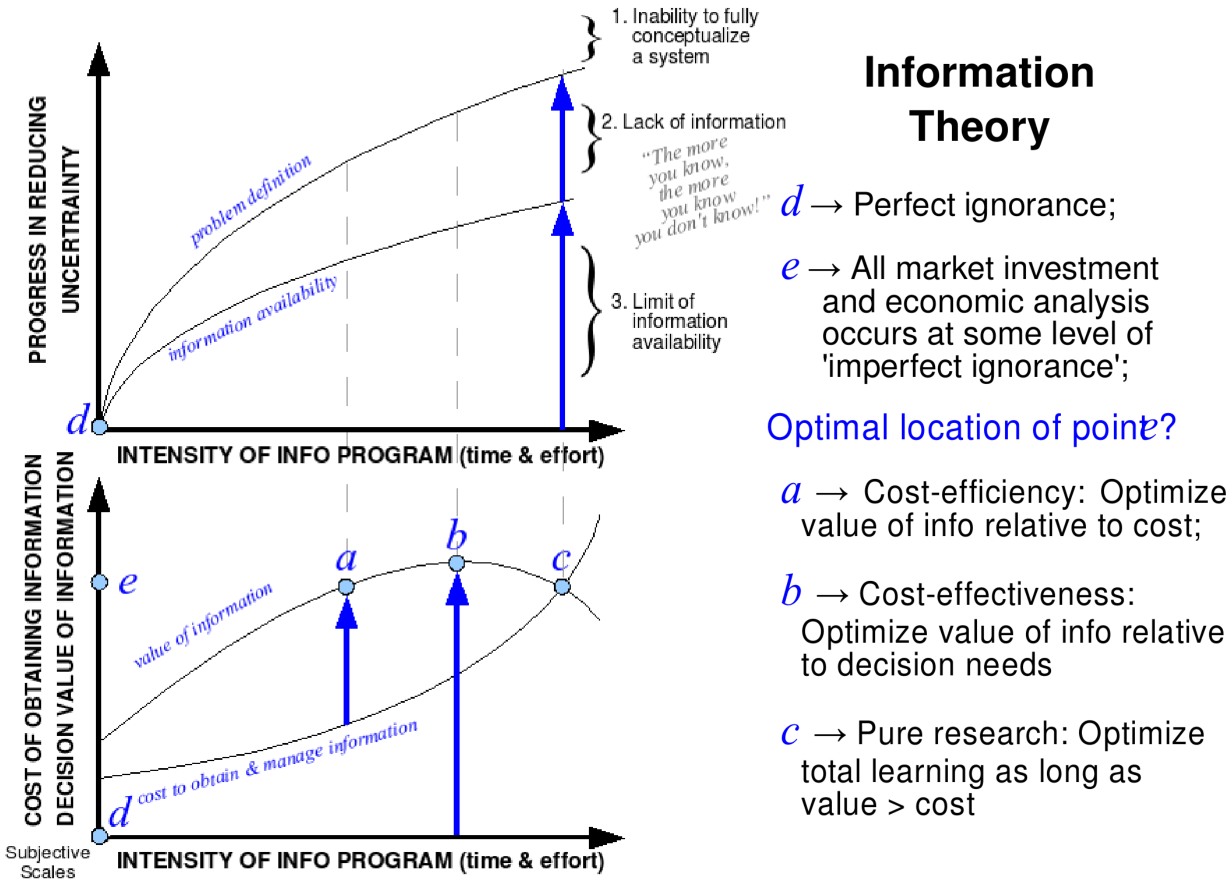
Readers without a background in physics or engineering may not appreciate that this involves more than an analogy. There is an extensive literature on the physical measure and practical utility of difference, or ‘gradients’, usually employing the term ‘*exergy*’. A recent review is provided by Tânia Sousa et. al. (Sousa et al., 2017)

This design research component of this work brings Edwin Jaynes’s critique of Claude Shannon’s information theory into practical relevance for project managers. The conceptual framework for this is adapted from Adrian McDonough’s Information Economics and Management Systems (McDonough, 1963).[13] The basic idea is summarized below.

Jaynes’ fixed point of 'perfect ignorance' is represented in Figure 5 at the origin of the *x* and *y* axes. This is a position that real project managers can relate to: a newcomer begins by knowing nothing about a problem. Similarly, an experienced manager begins by knowing nothing about a new problem.

13 I first adapted McDonough’s illustration for a project investment portfolio selection framework that I prepared under contract to the World Bank in 1992, for the Global Environment Facility (Potvin, 1992). It was enhanced again for use in a manager training course that I was contracted to design for a global Fortune 500 company in 2014, and that version of the diagram was then also published online for the Open Source Initiative (OSI). The explanatory text in this subsection is adapted from that most recent documentation I prepared for OSI (Potvin, 2014c).

Progress in reducing uncertainty is made by formulating questions and seeking answers in an iterative sequence of learning. In the upper half of the diagram, a problem can be incrementally conceptualized with increasing scope and precision through expanding investigation portrayed along the *x*-axis, proceeding further along the *y*-axis away from the state of maximum uncertainty at the origin. In response to questions raised in defining a problem, the amount of information that can be acquired increases with the intensity of learning effort. There is no upper limit to how much an observer could possibly know, but even so, what is learned typically arrives short of fully answering the known questions.



Source: POTVIN, J. 1992. Classification and Appraisal Criteria for Conservation Investments: A Proposed General Framework. Prepared under contract to The World Bank. Extended from: McDONOUGH, A. 1963. Information Economics and Management Systems. McGraw Hill.

Figure 5: A Pragmatic View of Uncertainty (Potvin, 1992), extended from a diagram by Adrian McDonough (McDonough, 1963)

Both curves are shown as horizontally hyperbolic to convey the view that greater and greater research yields no limit to how much a problem can be conceptualized or how much information can be generated about it. This representation also suggests decreasing returns to any particular line of research. The increasing gap illustrated between them merely incorporates the old adage: *the more you know, the more you know you don't know*.

In the lower portion of the main diagram, it is suggested that the cost of producing each additional amount of information tends to increase at a greater rate as more effort is directed to its acquisition. In contrast, the decision-making power of additional information, in terms of relevance, timeliness, accuracy and usefulness, is considered to increase only up to an optimum level, point '***b***', and then decline. Note that the 'value of information' refers here to its value 'in use' for a decision-maker considering a particular problem, and may not relate at all to value 'in exchange'.

In the pursuit of any objective, it may be imagined that there exists an optimal information base appropriate to each particular set of decision circumstances. This is depicted in the diagram as the level of information availability in the upper part of the diagram that corresponds with point '**b**' in the lower part. The curve in the lower part representing the value of information declines to the right of point '***b***' for two reasons. First, the longer the period of study before a decision is made, the less value can be attached to any information produced. Information today is worth more in any particular decision context than the same information later on. Too much of a delay before taking action, just to undertake more and more research, will even reduce the value of information that was previously generated. The second reason this curve declines past point '***b***' is that the decision function eventually becomes obfuscated by an oversupply of information. Joseph Levitt expresses this common problem:

> The more abundant the information, the less meaning it seems to yield. All seems, instead, congestion and confusion. The surest way to destroy a person's capacity for discrimination and good judgment is to bombard him or her with an enormous abundance of data, even if it's incontestably relevant. ... What is needed is discrimination in the supply and use of data, not their sheer abundance, regardless of relevance. ... Magnitudes must be limited to what is relevant and comfortably usable. The effective use of information is governed by the principle of parsimony: limit it to the more-or-less precise purpose at hand. A good thing is not necessarily improved by its multiplication. The governing question is: what is the question to be answered, the problem to be illuminated, the matter to be explored, the issue to be defined. And it is precisely because these are not self-defining concepts that it is essential to think them through in advance, because no amount of data will tell you what information you'll need to get at the right questions. (Levitt, 1991, p. 6)

The research effort therefore cannot be solely concerned with maximizing absolute decision power under a given set of decision circumstances. Simply put: the optimal information base may involve too much costs and delay. In the lower portion of our hypothetical illustration, maximum decision-making value per unit cost of information occurs at point '***a***', where the spread between the two curves is greatest. The additional study and considerable extra cost needed to develop an information base that maximizes decision-making power denoted by point '***b***' achieves, in the upper portion of the picture, only a comparatively minor reduction in uncertainty. Funds are limited, and it might not be worth the added cost.

On the other hand, the likelihood of error is greatest when information is evaluated solely according to its cost of production. That would lead to an information system being shaped according to the ease and cheapness of information recovery, rather than by decision priorities. Under those circumstances, statistics that are easiest and cheapest to recover would be given greatest attention, and those which are perhaps more subtle, costly and difficult to compile would tend to be ignored, even if they may be truly more valuable in the assessment of a problem. Since information about complex dynamic systems tends to be relatively more difficult and expensive to acquire, very valuable types of information could be systematically ignored.

The research effort should thus be designed to get the decision-maker somewhere between points '***a***' and '***b***', that is between the maximum decision-making value per unit cost of information, and the maximum absolute decision power. Whether the desired point is closer to '***a***' or to '***b***' depends upon the nature of risks associated with wrong decisions. The greater the risks, the closer we wish to get to point '***b***'. It is of great importance to stress again here, however, that generating too much information is still a mistake. Dangers are averted by getting the right amount of the right kind of information, at the right time, to people who can use it to make decisions. And it is always beneficial to push down the cost of obtaining and managing information, as discussed below.

3.4.2 Three Postulates for Optimal Rule Transmission Systems

What makes any particular rule system design better or worse than any other. Presumably the outcome of design research towards ‘optimal’ rule systems would be to enable individuals and entities to communicate normative propositions more *cost-efficiently* and *cost-effectively* than is otherwise currently feasible:

- *Cost Effectiveness:* Maximize the quality of direction-intrinsic data communication within a given amount of time, resources and risk.
- *Cost Efficiency:* Minimize the time, resources and risk needed to achieve an intended quality of direction-intrinsic data communication.

These *cost efficiency* and *cost effectiveness* criteria are found in an approximately equivalent form in all four of the leading international project manager competency standards (ISO, 2012, Section 4.3) (IPMA, 2006, Section 3.10, p 150-153) (Crawford & Duncan, 2007, p. 4) (ICCPM, 2012, Views 3, 6, 9). *Efficiency* and *effectiveness* goals may seem to be workable criteria, but they both depend upon an indeterminate notion of ‘quality’. For our present context we extend them with three postulates for rule systems design quality:

1. *<u>An optimal rule system</u> within a jurisdictional cluster of arbitrary individuals and entities is one that demands the least effort for them to categorize and communicate their respective normative propositions.*
2. *<u>An optimal rule</u> between any two or more individuals or entities is one that is centred upon their respective priorities, while also intersecting their shared points of agreement.*
3. *<u>An optimal set of rules</u> among multiple individuals or entities is one in which all the rules together reveal an emergent straight line of reasoning.*

These are stated as hypotheses to be tested in a later round of research involving field implementations and empirical results. For the present work these are only used as design guidelines. Appendix B describes in detail the metaphor and underlying approach to thinking about information theory that the present author employed to visualize these optimization postulates for rule systems, for sets of rules and for individual rules,[14] having prevously used it to structure several other distributed decentralized multi-entity initiatives throughout more than two decades.

3.4.3 Considering Rules Transmission Systems Technology in Light of Systems Ecology Theory

This research positions rule systems technology at the 'meso' level of a 'micro-meso-macro' framework combined with a 'project ecology' theory of agency structured in organizations, communities, networks and technologies. Shared concepts amongst economists, biologist, business strategist, ecosystem scientists and engineers is more than analogy or metaphor. These domains have enjoyed centuries of thoughtful discourse. Below we review some of the essential literature that ties these concepts together.

Technology was described as an evolutionary extension of human corporal and intellectual ability by the philosopher Ernst Kapp (Kapp, 1877). Also in the late 1800s, Volume 1, Chapter 1, Part 1 of Herbert Spencer's *Principles of Sociology* led off with the notion of "co-operation among sundry differentiated classes of individuals" leading to orderly action exhibited throughout communities "in their growths, structures, functions, products" (Spencer, 1898). What Spencer called "super organic evolution" is referred to by today's biophysical systems theorists as the "evolution of superorganisms" (Lovelock, 1993). This perception of technology as part of the evolutionary character of *homo sapiens* was further developed by bio-statistician Alfred Lotka (Lotka, 1925) and economic theorist Nicholas Georgescu-Roegen (Georgescu-Roegen, 1975) (Bobulescu, 2015), and the perspective was brought into general philosophy by Karl Popper:

> "But man instead of growing better eyes and ears, grows spectacles, microscopes, telescopes, telephones, and hearing aids. And instead of growing better memories and brains, we grow paper, pens, pencils, typewriters, dictaphones, the printing press and libraries." (Popper, 1979, pp. 238–239)

Bertram Brookes and Gregory Newby adopt this framework to describe computerized memory and information retrieval systems as evolutionary extensions to biological human memory (Newby, 2001, p. 1028). David Berry describes built infrastructures, such as transportation systems, or The Internet, as evolutionary extensions of human communities and societies (Berry, 2018) (Berry, 2019). Dermot Breslin et.al. further describe the "joint evolution of entities at multiple levels" (Breslin, Romano, & Percival, 2015) and Peter Bernus et.al. speak of a business ecosystem as "a system of organizations that co-evolve their capabilities and roles, and align their investments to create additional value, greater effectiveness and higher agility". These are not necessarily species-strengthening

14 Cognitive psychologist Frederic Bartlett introduced the term *schemata* a century ago to refer to sets of connected ideas and relationships in the human mind. (Bartlett, 1932) George Miller, Nelson Cowan and others explain the importance of organizing information to chunks. (Miller, 1994) (Sedig et al., 2005) (Mathy & Feldman, 2012)

developments, since at the individual level in the course of thousands of years “the need for intelligence was reduced as we began to live in supportive societies that made up for lapses of judgment or failures of comprehension”. (Crabtree, 2013b, p. 4) (Crabtree, 2013a) (Crabtree, 2013c) (Crabtree, 2013d) Does *homo sapiens* (wise) eventually cede to *homo callidus* (clever)?

The concept of “resilience” originated in social/institutional economist Joseph Schumpeter's notion of “creative destruction” which he first described while in the midst of the Second World War (Schumpeter, 1942) (Dopfer, 2012). The notion become central to the domain of ecosystem science beginning in the 1960s and over the decades it has come to have a range of meanings in multiple disciplines (Heckmann et al., 2015) (Smith et al., 2011) (Sterbenz et al., 2010). Following the global and regional market disruptions since 2007, the concept made a return trip to prominence in management literature, especially in reference to value-chain resilience. There are several recent literature reviews in that realm alone (Thomé et al., 2016) (Hohenstein et al., 2015) (Tukamuhabwa et al., 2015) (Santanu Mandal, 2014) (Bhamra et al., 2011). Every project evolves within a portfolio of projects (Killen et al., 2012) (Petit & Hobbs, 2010) (Müller et al., 2008) (Luehrman, 1998), within a supply chains (Fayezi et al., 2012) (Santanu Mandal, 2014) (Hohenstein et al., 2015) (Tukamuhabwa et al., 2015) and within markets (Rosa et al., 1999) (Çalışkan, 2009).

Recently Peter Bernus et.al. have described the business ecosystem as “a system of organizations that co-evolve their capabilities and roles, and align their investments to create additional value, greater effectiveness and higher agility”. They suggest that improvements at the project level typically require catalysts such as resilient inter-organizational interfaces, and an attractive incentive structure (Bernus et al., (In Press), p. 19) – essentially what was referred to earlier as the ‘meso level’ (rule systems):

> “The first challenge is to ... re-gain the coverage of the entire business on all levels of management, and a holistic and systemic coverage of the enterprise as an economic entity in its social and ecological environment. The second challenge is how to face the problems caused by complexity that limit the controllability and manageability of the enterprise as a system. The third challenge is connected with the complexity problem, and describes fundamental issues of sustainability and viability. Following from the third, the fourth challenge is to identify modes of survival for systems, and dynamic system architectures that evolve and are resilient to changes of the environment in which they live.” (Bernus et al., (In Press))

A general framework for conceptualizing rule systems as normative basis of projects is supplied by Kurt Dopfer, John Foster, and Jason Potts. They provide an interpretation of Joseph Schumpeter's three-level framework that distinguishes *micro* (behaviours of decision-makers amongst organizations), *meso* (rules, normative infrastructure and practices amongst industries and markets) and *macro* (characteristics of whole societies and economies) (Dopfer, 2012) (Dopfer et al., 2004) (Schumpeter & Boody, 1954). As illustrated in Figure 6 from their paper, they propose that efforts to achieve performance improvements across the micro level, where project managers operate, can be pursued most effectively at the meso level where normative infrastructure and practices of industries and markets are formulated and managed.

A *meso*-level design change that achieves widespread adoption at the *micro*-level can result in emergent and transformative *macro*-level effects for whole societies and economies. Whether those outcomes are 'better than', 'equivalent to', or 'worse than' the prior scenario or alternative outcomes rests upon a premise about which way is 'forward'.

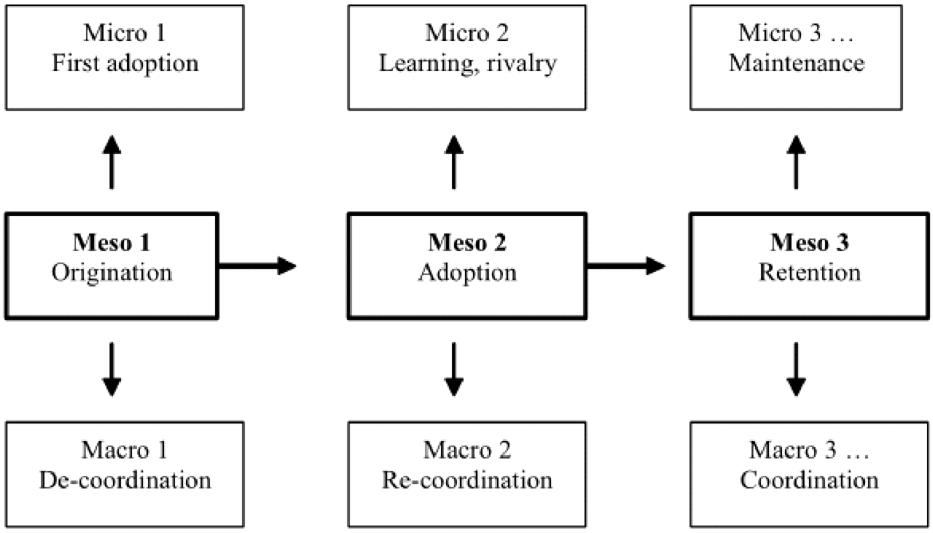


Figure 6: Micro-Meso-Macro Dynamics (Dopfer et al., 2004)

The meso level does not refer to anything external to the micro level project manager. Rather it refers to factors that are intrinsic to coordination amongst project managers and their stakeholders, and throughout the networks they participate in, such as the value chains they benefit from and contribute to, and the multi-layered jurisdictions they are members of. This "nested hierarchy" is independent of scale (Allen & Hoekstra, 1990, 8-9)

Austrian economic geographer Gernot Grabher introduced the phrase "project ecology" to the project management literature in 2004 to take account of "the interface between projects and the organizations, communities, and networks in and through which projects operate" (Grabher, 2004). This extension of the *micro-meso-macro* framework advocated in the *Oxford Handbook of Project Management* (Grabher & Ibert, 2011) (Grabher, 2004), and by one of the founders of the project management domain, Jonas Söderlund (Söderlund, 2004). It is also featured in two *Academy of Management Review* special issues (Bies et al., 2007) (Hitt et al., 2007). Project ecology frames a project on multiple levels : "from the micro-level of interpersonal networks to the meso-level of intra- and inter-organizational collaboration to the macro-level of wider institutional settings. Moreover, it unfolds a complex geography, which explicitly is not reduced to local clusters but also extends to more distanced individuals and organizations or a-spatial institutions” (Grabher & Ibert, 2011). A project is pursued amidst a plurality of relationships.

In a literature review published a decade ago, Söderlund observed that multi-firm multi-project studies involving institutional environments and rules to which the individual organization must respond and conform, had been given insufficient attention in prominent academic journals of project management (Söderlund, 2004, p. 483). There are several corresponding perspectives on this among authors who describe network platform innovation as a type of meso-level infrastructure project that involves the introduction of new methods, incentive mechanisms and business models (Kogut & Kulatilaka, 1994) (Fehrer et al., 2018) (Fu et al., 2018a) (Fu et al., 2018b). Peter Morris and Joana Geraldi have described institutional meso-level management activities as “creating a supportive institutional context” for micro-level agent interactions to flourish (Morris & Geraldi, 2011). David Berry describes general-purpose meso-level infrastructure such as legislation and courts, transportation systems, the Internet methods and equipment as evolutionary extensions of human communities (Berry, 2018) (Berry, 2019). Such systems of “shared agency” (Bratman, 2014) or “distributed agency” (Enfield & Kockelman, 2017) may be viewed at small scale (neighbourhood, municipal) or large scale (national, global) (Allen & Hoekstra, 1990, 8-9). While enhanced capabilities for individual organizations enable new “real options” for them (Driouchi & Bennett, 2012), an enhanced network platform opens up entire new “chains of real options" (Luehrman, 1998, p. 90), facilitating synergistic inter-dependencies for the networked entities through time (Archer & Ghasemzadeh, 1999) (Kapsali, 2011).

Therese Dille and Jonas Söderlund have observed that “project managers ... maneuver their projects in political, public and/or institutional environments” but that “conventional analysis of project organization has paid limited attention to projects in their institutional environment” (Dille & Söderlund, 2011, p. 480). In a literature review published a decade ago, Söderlund proposed a four-quadrant classification of project research by single-firm versus multi-firm studies, and single-project versus multi-project studies. This is shown in Figure 7.

Use of the term “ecology” in this context of projects and project managers is semantically precise. It is consistent with its particular meaning in the biophysical sciences, where it refers to the study of relationships amongst a plurality of self-organizing life forms (including humans) and with their physical environment (Holling, 1973) (Holling, 1986). Ecological analysis in relation to nature is not framed by geographical or temporal scale, but by comprehensiveness of view. Systems ecologist Tim Allen explains that when facing such a scenario, several different world views or perceptions are required to describe the system and to solve problems. No particular perspective is adequate by itself (Allen, 1987). His advice is equally pertinent to project ecology.

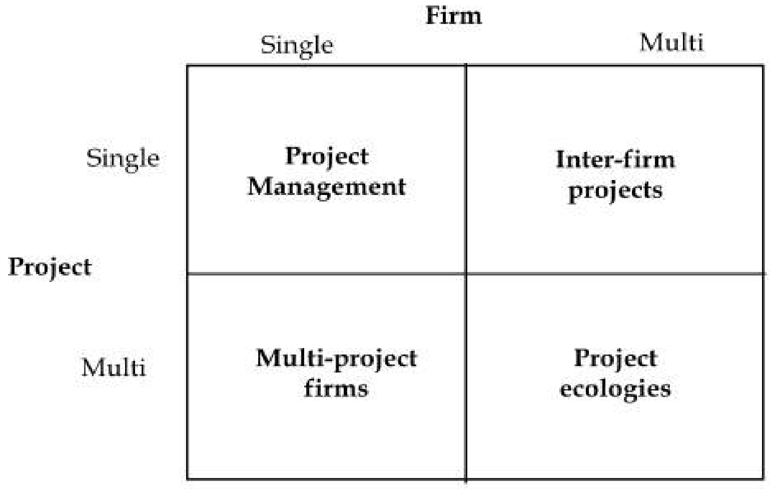


Figure 7: Categories of Project Management Research (Dille & Söderlund, 2011)

Systems theorist Robert Ulanowicz describes ecosystem theory as "mostly a relational endeavour" concerned more with the flows across many arcs than the characteristics of many nodes. "Such emphasis on relationships allows the discipline to be interpreted as the study of the communication of material, energy and information among systems components." (Ulanowicz, 2001, p. 393) He traces this perspective back to work by Raymond Lindeman in the 1940s, Robert MacArthur in the 1950s, then Eugene Odum in the 1960s.

Earlier we referred to David Berry's perception of built infrastructure, such as transportation systems and The Internet, as evolutionary extensions of human communities. Peter Morris and Joana Geraldi have described institutional-level management activities that occur "outside and around the project" to facilitate successful project development and delivery by "creating a supportive institutional context" for projects to flourish (Morris & Geraldi, 2011). They and other "neo-institutionalists" after John Meyer and Brian Rowan (Meyer & Rowan, 1977) see value in pursuing project management research at a more aggregated level. To this end, Darius Plikynas and Aistis Raudys also speak of a "theory of distributed agency" (Plikynas & Raudys, 2015). Direct pursuit of internal efficiency and effectiveness by individual project managers acting autonomously at the micro level does little to resolve macro-level problems (Kapsali, 2011, p. 405).

Meso-level projects to design and implement new or enhanced network platforms can change whole industries, markets and economies. Such changes could be desirable, or undesirable. Meso-level changes can extend or erode human corporal and intellectual ability and agency in the manner described by Ernst Kapp, Alfred Lotka, Nicholas Georgescu-Roegen, Karl Popper and Gerald Crabtree (Kapp, 1877) (Lotka, 1925) (Georgescu-Roegen, 1975) (Popper, 1979, p 238-239) (Crabtree, 2013 a,b,c,d). When new *meso-level* systems, such as modified network platforms, improve or degrade the tangible performance of individuals or organizations at the *micro-level*, this results in incremental behavioural changes which can eventually materialize into emergent and transformative *macro-level* effects for whole societies and economies.

Chapter 4: Review of Available Methods for Rule Logic Processing

4.1 Available Methods for Expressing Rule Logic

A rule communicated in natural language can also be expressed and considered in various other forms, such as in a computer programming language, a graphical flowchart (tree-diagram), and as an input/output table in the style of a 'logic table', 'truth table', 'control table', or 'decision table'. Each involve comparing a package of input data with a set of conditions, assumptions, triggers or premises, and the output data would be generated as instructions, replies, actions, options or conclusions.

In the following sections we step through a direct comparison of various ways of expressing a grocery store's delivery policy.[15] This begins with a short natural language policy as it may have been written by the manager of a grocery store. To keep this scenario simple, we may imagine a data source generated directly by an employee of the store, or by a self-check-out customer, who is putting the merchandise into a box. Once a box is packed, the person can select, or not select, each of three radio buttons on a touchscreen tablet:

❍ "Box is at least half full."

❍ "Box type is standard"

❍ "Value of box contents is greater than $100.00

Our purpose in this section is not to illustrate an entire workflow, but only the determination of whether or not delivery is offered, and if so, whether it would be free, or if there will be an additional fee.

It will be apparent in the examples below that RuleSpeak is a very simple and accessible form of rules-as-prose, which reduces rules to declarative statements that are concise, consistent and unambiguous (Ross, 2009). However RuleSpeak is not a language, schema or syntax. It is a guideline incorporating a set of 'best practices' conformant to the OMG's SBVR standard (Semantics of Business Vocabulary and Business Rules) (OMG, 2016b). The benefit of RuleSpeak is that its formal use of natural language for descriptive modelling or rules and logic can be easily understood by humans and machine-processed so long as the system has a suitable parser (Chapin, 2008). Anyone familiar with English can read and follow all three of these statements. On the other hand, there would still be need of some sophisticated natural language processing software to parse any of these to enable machine determination of which result is invoked in light of a circumstance.

15 Our example is adapted from an airline beverage service policy example by D. Robert Baker. (Baker, 2004)

The same rule can be expressed in computer programming code for machine determination, such as procedural *Ruby*, declarative *RuleML* and declarative *Notation3*. As will be illustrated with examples below, such rules-as-code methods contain expression and markup styles that are particular to each programming or markup language. Their unique styles require that when any of these expressions would be shared, recipients would require a particular line of software application that can parse the language-specific expressions and markup. Other recipients would need to rewrite them to operate on different application platforms.

It is feasible to implement rules-as-code for a particular application platform, and many commercial and government organizations do offer services in the real world based on this approach. However it is very labor-intensive, involving highly-trained personnel, and the resulting code is complicated, with the result that development and maintenance of the code library is very expensive. There are many 'successful' rules-as-code proof-of-concept projects, very few that are able to scale to tens or hundreds of thousands of rules due to the enormous volume of specialized work that would involve. Some rules-as-code projects are successful (Schneider, 2018) (République Française, 2016) and others devolve from small successful pilots into infamy as massive "incomprehensible failures". (OAG Canada, 2018)

Programming code cannot be re-located to other platforms, so all rules-as-code implementations need to be re-written and re-tested for porting to other applications. Following are illustrations of how a sample rule can be written using various methods. To ensure a fair comparison, each technical example below has been provided by someone who is experience in the method of expression.

4.1.1 Unstructured Natural Language Expression

We begin with the natural language expression of a store policy on delivery.

> When our standard delivery box is more than half full and also contains at least $100.00 in value of groceries, we provide free delivery. This does not apply to non-standard boxes. For all non-standard boxes, when delivery is provided we do charge for delivery. Delivery is offered for all boxes, standard or non-standard, whenever they are more than half full.

This could be more elegantly stated, but real policies are commonly stated in a slightly cumbersome way, such that one needs to read them slowly, more than once, to grasp their meaning. Following is a simpler, shorter natural language version rewritten with a language application. (Potvin & OpenAI/GPT-3, 2022)

> Delivery is available for all boxes, standard or non-standard, when they are more than half full. If our standard delivery box is more than half full, and it contains at least $100.00 in groceries, then delivery is free. For non-standard boxes, delivery is charged when it is provided.

4.1.2 RuleSpeak and SBVR

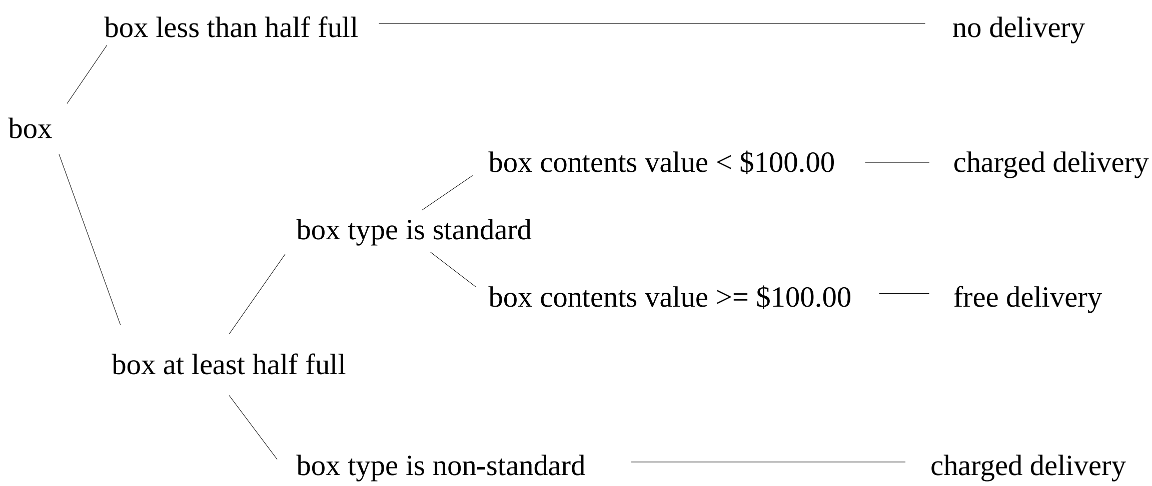
The Semantics of Business Vocabulary and Business Rules (SBVR) (OMG, 2016b) standard, which evolved from RuleSpeak, *(OMG, 2016c)* is designed to provide concise, consistent and unambiguous declarative sentences in natural language, which can be read as 'information' by a human directly from a user interface without any specialized expressions or markup. The following simplified expression of our sample rule was re-composed into SBVR/RuleSpeak-conformant expressions by Ron Ross, Co-Founder, Business Rules Solutions Inc. and principal designer of RuleSpeak, and lead author of SBVR. (Ross, 2022)

* A box may be delivered only if more than half full.
* The delivery of a non-standard box must incur a charge.
* The delivery of a standard box must be free if the box contains at least $100.00 of groceries.

4.1.3 Flowchart (Decision Tree)

This rule can be illustrated in a simple 'decision tree' style flow chart, as follows:

Figure 8: Flowchart Example



4.1.4 Procedural Imperative Programming Code e.g. PASCAL

The same rule was written in procedural programming language PASCAL *(Wirth, 1976)* by Don Kelly, full-stack developer, Production Engineering Group, Shopify. (Kelly, personal communication, May 18, 2020)

```
IF “box_at_least_half_full” = “T”
       THEN IF “box type is standard” = “F”
                     THEN actions = "delivery_is_available" + "delivery_is_charged"
              ELSE IF “box_contents_value” >= $100.00
                            THEN actions = "delivery_is_available"
                            ELSE actions = "delivery_is_available" + "delivery_is_charged"
                     ENDIF
              ENDIF
       ELSE actions = ""
ENDIF
```

This can be expressed similarly in generic pseudocode:

```
IF (box is more than half full)
THEN
    IF (box is non-standard)
    THEN (charge for delivery)
    ELSE IF (box is standard AND contains at least $100.00 of groceries)
    THEN (delivery is free)
    ENDIF
ENDIF
```

4.1.5 RuleML, OASIS

The identical sample rule was written in the RuleML mark-up language *(Boley, 2006)* by Theodoros Mitsikas, researcher at RuleML Inc. and PhD Candidate, National Technical University Athens. (Mitsikas, personal communication, April 12, 2022)

```
delivery(?x,no) :- box(?x),  naf(isHalfFull(?x)).

delivery(?x,charged) :- box(?x),  isHalfFull(?x), type(?x, standard), value(?x,?y), lessThan(?
y,100).

delivery(?x,free) :- box(?x),  isHalfFull(?x),  type(?x, standard), value(?x,?y), 
greaterThanOrEqual(?y,100).

delivery(?x,charged) :- box(?x),  isHalfFull(?x), type(?x, non_standard).

box(box_1).

 isHalfFull(box_1).

type(box_1,standard).

value(box_1,50).
```

4.1.6 Notation3

William Van Woensel. a post-doctoral research fellow at Dalhousie University Faculty of Computer Science (Van Woensel., personal communication, April 11, 2022) offered to express the same sample rule in Notation3 *(Berners-Lee & Connolly, 2008)*.

```
@prefix log: <http://www.w3.org/2000/10/swap/log#> .
@prefix math: <http://www.w3.org/2000/10/swap/math#> .
@prefix : <http://example.org/> .

:box1 :filled 49 ; :value 110 ; a :Standard .
:box2 :filled 70 ; :value 110 ; a :Standard .
:box3 :filled 70 ; :value 70 ; a :Standard .
:box4 :filled 70 ; :value 110 ; a :NonStandard .

{ ?box :filled ?level . ?level math:notLessThan 50 } => { ?box :delivery [ :available true ] } .

{    ?box :delivery ?delivery . ?delivery :available true .
     ?box :value ?value . ?value math:lessThan 100  .
} => { ?delivery :charged true } .

{    ?box :delivery ?delivery . ?delivery :available true .
     ?box a :NonStandard
} => { ?delivery :charged true } .

{    ?box :delivery ?delivery .
     _:x log:notIncludes { ?delivery :charged true } .
} => { ?delivery :free true } .
```

4.1.7 Decision Model and Notation (DMN), OMG

Expression of our sample rule was provide in Decision Modeling Notation (DMN) (OMG, 2019a) by Bernd Rücker, co-founder of Camunda Inc. which supplies a free/libre/open source platform for BPMN and DMN decision automation. (Rücker, personal communication, 28 May, 2020)

Figure 9: DMN Example

Delivery Offered

delivery

View DRD

F	Input +			Output +	
	Box Type	Grocery Order Sum	Fillig Degree (%)	Delivery Status	
	string	long	long	string	Annotation
1	"Standard"	> 100	> 50	"free"	
2	"Standard"	-	> 50	"charged"	
3	"Standard"	-	-	"not offered"	
4	"Non-Standard"	-	> 50	"charged"	
5	"Non-Standard"	-	-	"not offered"	
+	-	-	-		

4.1.8 Truth Table (Input/Output Binary Decision Table)

Below is a method of expressing the sample rule in a simple vertical input/output 'truth table' ("vertical I/O table"), adapted from a general structure implemented in the tableBASE rules engine by DataKinetics Inc., and described by Wayne Cunneyworth (Cunneyworth, 1994). Expressing rules this way is machine processable if it is standardized, and applications are programmed for what to do with such data.

Table 8: Truth Table Example

INPUT DATA	Scenario A	Scenario B	Scenario C	Scenario D
box_at_least_half_full	F	T	T	T
box_type_is_standard	-	F	T	T
box_contents_value >= $100.00	-	-	F	T
OUTPUT DATA				
delivery_is_available		X	X	X
delivery_is_charged		X	X	

By convention:

- INPUT state is "T"=present; "F"=absent; " - "=has no effect on the outcome (also referred to as "don't care").
- OUTPUT state "X"= is confirmed; " "= is not confirmed (blank)

Illustrating the logic in this way has the significant advantage that one can readily see that the Scenarios B and C, while containing different T and F permutations, are nevertheless associated with the same output. Thus an analyst or an algorithm can consolidate these two scenarios into a single scenario. By convention, T and F values which have no effect on the outcome are replaced with a hyphen. (Cunneyworth, 1994, p, 55) The consolidated Table 9 for this example has three scenarios, as two with a common output are merged as Scenario B. The data formerly listed as Scenario D now appears as Scenario C.

Table 9: Consolidated Truth Table Example

INPUT DATA	Scenario A	Scenario B	Scenario C
box_at_least_half_full	F	T	T
box_type_is_standard	-	-	T
box_contents_value >= $100.00	-	F	T
OUTPUT DATA			
delivery_is_available		X	X
delivery_is_charged		X	

By convention:

- INPUT state is "T"=present; "F"=absent; " - "=has no effect on the outcome (also referred to as "don't care").
- OUTPUT state "X"= is confirmed; " "= is not confirmed (blank)

4.2 Alternative Logic Data Models

4.2.1 Limitations When Comparing Logic Models

This section considers various sets of elements employed in different logic frameworks. Magdalena Pradilla Rueda, an historian of logic frameworks advises that throughout the evolution of formal logic domains—*essence and being; language of reasoning; procedure from premises to conclusions; constructs of mathematics; and, explanations of tangible phenomena*—each set of concepts and methods can only be understood in its respective context of a particular theory of knowledge and its corresponding philosophical system. (Pradilla Rueda, 2017, p. 143-144)

Among constructs of mathematics, each logic data model occurs in what is referred to as a Hilbert space.[16] Hilbert spaces are conceptual models that involve the coupling of axioms from two or more different domains of mathematics, abstracted from their *semantic* content. To achieve this coupling involves carefully establishing mathematical *syntactic* correspondence among the axioms of each domain. By applying Hilbert's principles of rigorous proof, one can ensure that theorems from the different domains do not contradict each other. (Pradilla Rueda, 2017, p. 118-124) However our method of expression in this section remains in the style of plain language, to prioritize reflection on the practical meaning of the concepts.

The commonly used "truth table" is just one of many variants available for the tabular expression of logical relations. This section reports on a review various logic data models undertaken in order to consider ideas that would be meaningful for general purpose normative logic, equally usable by humans

16 Named after the mathematician David Hilbert, a "Hilbert space" is a set of general relations spanning two or more fields of mathematics (arithmetic, algebra, geometry, trigonometry, calculus, number theory, combinatorics and probability). It is a meta-relation of mathematical theory that is used for combining principles across these basic mathematical domains into a carefully-coupled set of principles in common, thereby providing useful ways to understand, predict and analyze how different shapes, numbers, and properties relate or interact. The term "space" is used in the sense of a conceptual space that contains the set of all possible solutions to a given problem type, so that any instance of that problem type can be explored and analyzed with the confidence that if a solution is possible, then it can be determined with the concepts that comprise that Hilbert space. However, as Kurt Gödel demonstrated, within any Hilbert space there may also exist propositions that cannot be proven to be either true or false. (This summary was simplified with (Potvin & OpenAI/GPT-3, 2022).)

and machines. The following few sections summarize various three-value and four-value logic data models, which through incremental consideration, informed my own novel design of a tabular structure suited to the objectives of the present undertaking.

4.2.2 Various Trinary Logic Models

Designers of computational systems have predominantly represented logic information with bits having two states {0,1}, beginning with the introduction of electronic binary logic for this purpose in the late 1930s by John Atanasoff and Clifford Berry (Gustafson, 2000). They adapted the idea of physical binary data representation from the operational method of holes in cards invented by Joseph Marie Jacquard for an automated mechanical loom, first produced in 1805 (Davis & Davis, 2005) (Delve, 2007). Trinary logic for numeric mechanical calculation was introduced just 25 years later by Thomas Fowler[17]. This facilitated simpler expressions and greater efficiency than was achievable with decimal or binary methods (Glusker et al., 2005).

Given the increased data density and processing speed achievable with a trinary data model, several variants have been designed and employed for distinct types of logic processing (Connelly, 2008), including Fowler's trinary designs (Glusker et al., 2005) (Khalid & Singh, 2016, p. 399), quantum methods (Nielsen & Chuang, 2010),[18] Jan Łukasiewicz's multi-valued structure (Łukasiewicz, 1920), Lotfi Zadeh's 'fuzzy logic' that accommodates a continuous analog gradient between the absolutes 0 and 1 (Zadeh, 1965), Emil Post's "true; false; incompletely-false-hence-true-enough" model (Post, 1921, p. 167-181), and Sylvain Hallé's test suite logic model. (Hallé, 2022)

- {-,○,+} — Fowler's Symbolic Symmetric Trinary Analytics
- {-1,0,1} — Fowler's Numeric Symmetric Trinary Analytics
- {0,1,2} — Fowler's Numeric Asymmetric Trinary Analytics
- {0,ψ,1} — Quantum Numeric/Probabilistic Binary with Asymmetric Trinary Analytics
- {0, *p*, 1} — Łukasiewicz's Binary with Multi-Valued Asymmetric Trinary Analytics
- {0, *x*, 1} — Zadeh's Fuzzy (Continuous Gradient) Binary with Asymmetric Trinary Analytics
- {⊤, ⊥, ?} — Hallé's True-False-Unknown Binary with Asymmetric Trinary Analytics
- {T,F,T} — Post's True-False-Incompletely False Binary with Asymmetric Trinary Analytics
- {T,F,N} — True-False-Neither Binary Asymmetric Trinary Decision Table
- {T,F,-} — True-False-Inconsequential Asymmetric Trinary Decision Table

17 Fowler's particular requirement was to solve the unwieldy task of balancing local tax revenues with unemployment insurance benefits under the complicated English pre-decimal currency system of 4 farthings to the penny, 12 pennies to the shilling, and 20 shillings to the pound. (Glusker et.al., 2005)

18 The basic ideas of quantum logic are well-understood, but quantum computing has not yet been accomplished beyond limited lab experiments using specialized hardware such as superconducting circuits, ion traps, photonic systems, and nuclear magnetic resonance, and its realization remains contingent upon as-yet undiscovered realities and uninvented methods. (Gomes, 2018) (Savchuk & Fesenko, 2019).

The choice between numbers or letters is significant, as logic tables with {0,1} or {-1, 0, 1} are precise, whereas there is an intrinsic ambiguity to systems that employ {T,F}. ‘True’ and ‘false’ may be interpreted in a variety of ways, and so-called ‘truth tables’ have multiple origins, styles and meanings for particular contexts (Anellis, 2012) (Shosky, 1997).[19]

The meaning of the third symbol of the trinary logic systems differs widely. For example:

- The ‘hyphen’ in {T,F,-} has a somewhat consistent meaning among many sources. Lew, Parnas and Cunneyworth all signal that the computational interpreter may ignore a particular data input field value because a given input element is *‘Of No Consequence’* to the specified output ‘X’ (Lew, 1983). This is variously referred to as an *‘Inconsequential’* or *‘Don’t Care’* signifier.
- The ‘N’ in {T,F,N} can have diverse meanings. This can be interpreted in English as ‘Neither’, ‘Null’, ‘Noise’ or even ‘Notify’ (i.e. a signal to flag a data quality issue).
- The Greek symbol Psi ‘ψ’ in quantum logic describes the probability of a particle being in one of two states at a given time, {0, ψ, 1}. Quantum-mechanical particles called qubits” store and process data, and the two states are assigned meaning for use in quantum logic. The qubit state |ψ> is a probabilistic superposition, not intended to express a fully-concurrent state of ‘0’ and ‘1’.[20] (Nielsen & Chuang, 2010, p. 13-14)

What is common among all of the ‘third elements’ of the trinary logic frameworks referred to above is that they are rationally definable. This is because three-element logic models accommodate *gaps* where referent statements may be less than entirely true or false, or as in the case of the ‘hyphen’, their status is irrelevant. In the following sections we consider various four-element logic models that incorporate the superposition of true and false, which Graham Priest has described as a truth value *glut* (Priest, 2018, p. 70), and Nuel Belnap et al. have described as “too much (inconsistent) information” (Anderson & Belnap, 1975, p. 512), which I categorize as enigmatic, Socratic and pragmatic contradictions.

19 A quarter millennium ago in his *Dictionary of the English Language,* Samuel Johnson devoted half a page to the meaning of ‘true’, stepping through eight semantic distinctions for the term, and he dedicated more than a page to the meaning of ‘false’ and its various derivative terms such as ‘falsifiability’ (Johnson, 1755). Bas van Fraassen has generally distinguished two epistemic attitudes towards the adjective ‘true’. *Scientific realism* pursues truth as literal fidelity between propositions and Nature; whereas *constructive empiricism* pursues truth as the empirical adequacy of propositions for various Human purposes (Van Fraassen, 1980).

20 This corresponds to Claude Shannon’s joint entropy *H(X,Y)* in formal information theory (Shannon, 1948), and similarly reflects Jakob Bernoulli's “Principle of Insufficient Reason” (Shafer, 1996) and John Maynard Keynes’ “Principle of Indifference” (Keynes, 1921). This duality appears in the archaic colloquial English term ‘habnab’, which means "let it happen or not; without any rule or certainty of effect" (Johnson, 1755). This element is the conceptual basis of undertainty in information theory (Tribus, Shannon & Evans, 1966) (Jaynes, 1979) (Eriksson et al., 1987) and thus it underlies comparative quantitative analysis of the performance and size of various decision systems relative to a fixed notional point of origin characterized by no rules. The simple illustration in the text that uses 'raining' and 'sunny's states can be adapted to Kurt Gödel's framework for logic (Gödel, 1931) as follows: "Quantum-mechanical particles called qubits” store and process data, which have two states {0, ψ, 1}, where the Greek symbol Psi ‘ψ’ describes the probability of a particle being in a particular state at a given time. These two states can then be assigned meaning for use in quantum logic. Let's imagine that a statement can be "true AND false" at the same time. If '0' is assigned the meaning “false”, and '1' is assigned the meaning “true”, then the statement “0 AND 1” can be represented by the qubit state |ψ>. An inventive logician wanting employ a quantum logic framework to forecast the chance of undecidability as defined by Gödel, could communicate the qubit state |ψ> representing the probability that a statement is both false and true for a given circumstance, thus causing an undecidable outcome."

4.2.3 Enigmatic Contradiction in Tetralemma Logic

Ancient Buddhist *tetralemma* (meaning *four proposition,* also known as *catuṣkoṭi*) metaphysics reaches back to Sañjaya Belatthiputta, the venerable teacher of the 6th century BCE on the Indian subcontinent, contemporaneous with the Buddha. He is reported as saying: “Were you to ask me whether there exists another world after death, well if I thought that there did exist another world, would I declare that to you? I don't think so. I don't think in that way. And yet I don't think otherwise. I don't think not, and I don't think not not. Should you ask me if there isn't another world, perhaps there both is and isn't, and perhaps there neither is, nor isn't.” (Adapted from various sources.) Somewhat later, in the Mādhyamika school of the the 5th century BCE, this took on a more formal expression as: {χ exists; χ does not exist; χ exists and does not exist; χ neither exists nor does not exist} (Staal, 1975, p. 57-58) Ranil Dion Guneratne has explained this *tetralemma* framework as grounded in the logic system of Indian Mahāyāna Buddhist philosopher Nāgārjuna (c. 150 – c. 250 CE), with the four states: {True, False, Both and Neither} or {T,F,B,N}. (Gunaratne, 1980) (Gunaratne, 1986) (Madanayake et al., 2015). *Tetralemma* (*catuṣkoṭi*) logic was brought into the Western curriculum by Graham Priest (Priest, 2006) (Priest, 2010) (Priest, 2014) (Priest, 2018), and was provided a formal logical vocabulary by Takuro Onishi (Onishi, 2015) and a generalized notation by mathematician Giuseppe Greco et. al. (Greco et al., 2019).

Unresolvable paradox is not foreign to Western philosophy, however. Bertrand Russell famously wrote: “There is a proposition which I am asserting and which is false.” (Russell, 1919, p. 356) – commonly referred to as “Russell’s Paradox” (Lawvere, 1969) (Linsky, 2002) (Yanofsky, 2003) (Studd, 2019). In 1931 Ernst Gödel provided a general conceptual foundation for undecidable propositions (Gödel, 1931). Logical superposition may be expressed as ‘True-AND-False’, ‘0 AND 1’, or ‘Yes-AND-No’. The physical sciences accommodate some persistent contradictions (Yanofsky, 2003) and Richard Feynman also referred to the “hidden-variable problem” (Feynman, 1982, p. 476), which he explained are factors that are unanticipated, misunderstood, unmeasurable, or otherwise unknowable. Even in applied mathematics, engineers have long reconciled with conceptually irrational numbers such as Pi, Euler’s *e*, and Gauss’ square root of -1.

4.2.4 Socratic Contradiction in Four-Valued Logic

Some contradictions are not unresolvable. These are systematically resolvable through refined hypotheses, more precise terms and definitions,[21] better methodologies and models, more data, deeper conceptual

21 The question: “*Is a bicycle tire a solid object?*” does not raise a Yes-AND-No enigma; but it does prompt a Socratic Yes-AND-No contradiction that can be resolved by replacing vagueness with precision. Nevertheless, until it is resolved with a more precise question, a Yes-AND-No contradictory answer is the most correct response.

understanding, and more time. In the Socratic Method, the purpose of contradiction is to challenge beliefs and assumptions, and to force critic thinking. Not all such contradictions need to be resolved, and even on important matters, John Maynard Keynes recognized a century ago that sometimes decisions need to be made under uncertainty:

> "It is difficult to see, however, to what point the strengthening of an argument's weight by increasing the evidence ought to be pushed. ... There clearly comes a point when it is no longer worth while to spend trouble, before acting, in the acquisition of further information, and there is no evident principle by which to determine how far we ought to carry our maxim about strengthening the weight of our argument." (Keynes 1921:77).

Jan Łukasiewicz introduced 'multi-valued' logic in 1920 (Łukasiewicz, 1920), which was refined by Belnap et al. as 'four-valued logic' to accommodate uncertainty arising from contradictory or missing information.[22] (Anderson & Belnap, 1975, 491-521) They articulated a very specific rationale for incorporating contradiction into the state variables of computing systems, and a relatively long excerpt is warranted for clarity on this key point:

> [We consider] "the computer as part of a network of nodes with which it exchanges information, where there is no single infallible source of the computer's data, especially where inputs come from several sources. ... In such circumstances *the* crucial feature of the situation emerges: *inconsistency threatens*. Elizabeth tells the computer that the Pirates won the Series in 1971; Sam tells it otherwise. What is the computer to do? If it is a classical two-valued logician, it must give up ...
>
> Of course we want the computer to report any contradictions that it finds, and in that sense we by no means want the computer to ignore contradictions. It is just that where there is a possibility of inconsistency, we want to set things up so that the computer can continue reasoning in a sensible manner even if there is such an in consistency, discovered or not. ...
>
> What we are proposing is to Keep our Data Clean. ... So we have a *practical* motive for dealing with situations in which the computer may be told both that the thing is true and that it is false (at the same time, in the same place, in the same respect, etc., etc., etc.). ... In the meantime, while others work on this extremely important problem, our computer can only accept and report on contradictions without divesting itself of them. ... the computer is to reply strictly in terms of what it has been told, *not* in terms of what it could be programmed to believe. For example, it if has been told that the Pirates won and did not win in 1971, it is to so report, even though we could of course program it to recognize the falsity of such a report. The point here is both subtle and obvious: if the computer would not report out contradictions in answer to our questions, we would have no way of knowing that its data harbored contradictory information. ...
>
> For each item in its basic data file, the computer is going to have it marked in one of the following four ways: (1) just the 'told True' sign, indicating that item has been asserted to the computer without ever having been denied; (2) just the value 'told False,' which indicates that the item has been denied but never asserted; (3) not 'told values at all, which means the computer is in ignorance, has been told nothing; (4) the interesting case: the item is marked with both 'told True' and 'told False.' " (Anderson & Belnap, 1975, 507-510)

Socratic contradiction may also persist due to people sticking to their different perspectives. Although Aristotle's long-standing "Principle of Non-Contradiction" holds that two contradictory statements cannot

22 We refer to "the Belnap et al. model" for reasons explained in this note by Michael Dunn:

> "So Belnap first created the formal logic of first-degree entailments. Smiley provided the interpretation of it in terms of an abstract 4-valued matrix. And Dunn first provided the intuitive interpretation of it. Therefore it should be called the "Belnap-Smiley-Dunn Four-valued Logic." But we should not just stop here without acknowledging that ideas suggesting the 4-values (True, False, Both, None) date back to classical Indian logic (Sanjaya's "Four Corners"), prior to the 6th century B.C.E. ... So maybe it should be called the "Sanjaya-Belnap-Smiley-Dunn Four-valued Logic." (Dunn, 2019, p. 21)

both be true at the same time, even he accepted that opposing views can be simultaneously correct in different ways, or that they can both have some truth to them, even if one is more accurate than the other. In his essay *Politics* Aristotle wrote: “at the same time it is not difficult to see that those who assert the opposite are also right in a manner.” (Aristotle, transl. Rackham, 1932, sec 1252a)

4.2.5 Pragmatic Contradiction in Tetranary Logic

Some contradictions are deliberately designed to persist, or tolerated in real-world law, business and operations. (Fletcher, 1985) (Cook, 1924) In the aptly titled collection of essays *‘The Reasonable as Rational?’* von Wright explained:

> “Some legal philosophers ... have thought that a legal order of necessity is ("has to be") gapless and contradiction-free. That this simply is not the case with (most) legal orders has been emphatically pointed out by Alchourrón and Bulygin ... Alchourrón's and Bulygin's criticism soon convinced me that a descriptive reading of the formulas of the standard system gives us a formal system of norm-propositions which cannot claim universal and necessary validity for its formulas.” (Von Wright, 2000, p. 174-175)

Von Wright accommodates indeterminacy through AND and OR:

> “Suppose there is in a given normative code no norm permitting, but also no norm prohibiting, either the state of affairs ρ or its contradictory ~ p. There is thus a ‘gap’ in the code. It can be abolished (‘covered up’) in one of two ways, viz. either by *permitting this state <u>and</u> its contradictory* or by *prohibiting this state <u>or</u> its contradictory*. If the law-giver agrees that the law should be without gaps, he will have to perform either one of these "legislative acts". But no norm-logical considerations can tell him which alternative to choose. (Von Wright, 2000, p. 177-178. Emphasis added)

Following are two explicit examples from law.

(a) Did I enter another country when I crossed the 49^{th} parallel?

At the time the 1814 *Treaty of Ghent* was signed to settle the War of 1812 between U.S. and the United Kingdom, the available geographical surveying methods were not as accurate as would have been required to position the international border between Washington State USA and the province of British Columbia, Canada at exactly the 49th parallel. The Treaty of Ghent remains ‘in effect’ today, and it states:

> “It is further agreed between the two contracting parties that ... all grants of land made previous to the commencement of the war by the party having had such possession, shall be as valid as if such Island or Islands had by such decision or decisions been adjudged to be within the dominions of the party having had such possession." (U.S. Senate, 1814)

More precise surveys performed under the subsequent 1908 *Treaty Between the United States of America and the United Kingdom Concerning the Boundary Between the United States and the Dominion of Canada from the Atlantic Ocean to the Pacific Ocean* identified some small areas of land that had previously been misallocated to one or the other country. However both countries are aware that to re-open negotiation over the boundaries established by the 1814 treaty would also bring significant parts of southern Ontario, Quebec and northern Minnesota, New York, Vermont, New Hampshire, and Maine into re-negotiation. Since neither country wants to do this, there are some small areas of land where authorities of both countries, while administering their respective portions as delineated in 1908, also allow each others' citizens to cross back and forth over the 49th parallel into the portions that they administer under this mutual arrangement. So long as these visitors remain within the demarcated areas that were misallocated in 1814, they are deemed to not have entered into the other country. (Brend, 2020)

To validate this, when I was in Surrey, British Columbia, Canada on 5 November 2020, during which time the border between Canada and the United States was strictly closed to all non-essential travel (Scherer, 2020), I consulted with a Canadian federal police officer who was patrolling this particular border area. In full conformance with what he said, I then walked from 0 Avenue, Surrey, about 30 meters south across the 49th parallel. I ate lunch at a picnic table in a United States administered park, and returned without incident in view of the same police officer. Did I enter the United States on that date? "Yes-AND-No".

(b) "Is a bumblebee a fish under California law?"

The California Court of Appeal released a determination on 31 May 2022, that for purposes of Section 45 of the California Endangered Species Act, the operational definition of 'fish' was deemed to include bumblebees:

> "The issue presented here is whether the bumble bee, a terrestrial invertebrate, falls within the definition of fish, ... Although the term fish is colloquially and commonly understood to refer to aquatic species, the term of art employed by the Legislature in the definition of fish in section 45 is not so limited. ... The legislative history supports the liberal interpretation of the Act (the lens through which we are required to construe the Act) that the Commission may list any invertebrate as an endangered or threatened species. ... We conclude a liberal interpretation of the Act,12 supported by the legislative history and the express language in section 2067 that a terrestrial mollusk and invertebrate is a threatened species (express language we cannot ignore), is that fish defined in section 45, as a term of art, is not limited solely to aquatic species. Accordingly, a terrestrial invertebrate, like each of the four bumble bee species, may be listed as an endangered or threatened species under the Act." (California Court of Appeal, 2022)

If one were to ask: "Is a bumblebee a fish within the meaning of California's Endangered Species Act?", the correct answer would be "Yes". However if one were to ask more broadly: "Is a bumblebee a fish under California law?", the correct answer would be "Yes-AND-No". Although a bumblebee "falls within the definition of fish" for the purposes of protecting endangered species in California at this time, one need not obtain a fishing license to catch a bee.

4.2.6 Distinguishing Among Various Two- and Four-Element Logic Models

Given the variety of meanings among various four-element logic models, it is useful to summarize their differences. In this dissertation the phrase ***"Tetranary Logic"*** [23] is used to refer generally to all types of four-element logic models, versus two-element (and three-element) ***"Binary Logic"***.

- ***Tetralemma Logic.*** Since at least the 6th Century BCE, "*tetralemma*" (*catuṣkoṭi*) logic has posited four states: {χ exists; χ does not exist; χ exists and does not exist; χ neither exists nor does not exist}. These are commonly summarized as {True, False, Both, Neither} (Priest, 2010);
- ***Normative & Deontic Logic.*** Since the 1950s *"normative and deontic logic"* has developed with the set {Obligatory, Permitted, Forbidden, Indifferent}, summarized as {OA, PA, ~PA, PA&P~A}. (Von Wright, 1951) [24] The IETF standard replaces the fourth term's neutrality with one of encouragement (Bradner, 1997) {MUST, MAY, NOT, SHOULD}, used in the DWDS model.
- ***Four-Valued Logic.*** Since the 1970s century the phrase "*four-valued logic*" has referred specifically to {told True; told False; told True also told False; untold True also untold False}. These can be summarized as {True, False, Both, Untold}.[25] (Anderson & Belnap, 1975);
- ***Nucleobase Logic.*** Since the 1990s genetic nucleobases adenine (A), cytosine (C), guanine (G), thymine (T), and uracil (U) have been harnessed for logic processing, using {A,G,C,T} for DNA storage and {A,C,G,U} for RNA processing. (Adleman, 1994) In some recent work the binary numbers {00, 01, 10, 11} were assigned to the nucleotides {A, G, C, T}. (Liu et al., 2022, p. 905)
- ***Quaternary Logic.*** Four-element quantitative numeric logic models are referred to as *"quaternary logic"* in this dissertation.[26] In electronic signaling for nano-devices {0,1,2,3} are 'thirds of a volt': {$^0/_3$V, $^1/_3$V, $^2/_3$V, $^3/_3$V}. (Moaiyeri et al., 2012). In 'cellular automata computing' for big data, {1,2,3,4} represent 'fitness scores' to iteratively prioritize solutions until one remains (Navidi et al., 2021) In 'reversible computing' for multiple solution scenarios, partial steps are given fitness scores so that computation continues only on those with the highest scores, while the rest are reversed. (Khan, 2008) The set {1,2,3,4} could also be used to indicate lines, planes, volumes and time-space fabrics in a logic framework of dimensions.

23 The term "tetranary" is borrowed from the field of chemistry, where it refers to compounds containing atoms of four different elements. This term was also chosen for a similar reason by Harrie van der Haghen, a PhD candidate at Wageningen University in the Netherlands with the schema: {No, Yes, Yes-AND-No, Yes-OR-No}:

> "Tetranary Logic for Self-Programming Software or Hardware.: ... I could have called this 'Quaternary logic' but this term is already occupied for calculation with two bits, boolean operations with two bits at a time, and interfacing integrated circuits with four different stages. The logic that I want to present here is not meant for quantitative operations, but for programming of logic in a learning process." (van der Haghen, 2016)

24 George von Wright proposed two deontic operators, in positive and negative states (emphasis added):

> "The proposition that the act named by A is **permitted** will be expressed in symbols by **P**A. The proposition that the act named by A is **forbidden**, is the negation of the proposition that it is permitted. It can thus be symbolized by **~(P**A). The proposition that the act named by A is **obligatory**, is the negation of the proposition that the negation of the act is permitted. It can thus be symbolized by ~(P~A). We shall also use the shorter expression **O**A. The proposition that the act named by A is (morally) **indifferent** can be symbolized by (PA)&(P~A). The proposition that the acts named by A and by B are (morally) **incompatible** can be symbolized by ~(P A&B). The proposition that the performance of the act named by A **commits** us to perform the act named by B can be symbolized by OA→B. But OA→B means the same as ~(P ~(A→B)) and this means the same as ~(P A&~B). Commitment can thus be explained in terms of compatibility. P and O are called the 'deontic operators'." (Von Wright, 1951, p. 4)

25 Belnap et al. actually summarize their model with {True, False, Both, None}, but reasons explained later in this section, this dissertation uses 'Untold' instead of 'None'.

26 Until this point in our consideration of logic data models, we have made no distinction between the use of numeric digits as numbers in their quantitative sense, versus the use of numbers merely as convenient symbols with other assigned meanings (i.e. 0 assigned to mean 'False' or 'No', and 1 is assigned to mean 'True' or 'Yes').

Various binary and tetranary logic data models are structured in Table 10 for direct comparison.[27] Keeping in mind the caution of logic theorist Magdalena Pradilla Rueda that each logic model emerges from particular philosophical roots, specialists in formal logic theory may notice the limitations and faults of the comparison offered here. The present purpose is not intended to achieve comprehensive scope or precision, rather it is only to respectfully situate within the field of formal logic my own 'DWDS' design, which is to be explained in subsequent chapters.

Table 10: A brief plain-language comparison of various logic data models used in computation.

Logic Model	**Type**	**States**	**Meanings**			
Binary Logic (can be used to simulate Tetranary Logic)						
Aristotelian Logic	Exclusive Binary	*False, True, Either, Neither*	False	True	Either 100% True OR 100% False	Neither T NOR F
Quantum Logic	Probabilistic Binary	*No, Yes, Expect Either, Neither* *or* *False, True, Expect Either, Neither*	No *or* False	Yes *or* True	Expectation 100%Yes/True with x% probability OR 100% No/False with y% probability, where x+y=100	Neither Y NOR N T NOR F
Fuzzy Logic	Proportionate Binary	*No, Yes, Proportion, Neither* *or* *False, True, Proportion, Neither*	No *or* False	Yes *or* True	Proportion x% Yes AND y% No, where x+y=100	Neither Y NOR N T NOR F
Tetranary Logic (can use Binary Logic as a subset)						
Tetralemma Logic (Catuşkoţi)	Metaphysical Tetranary	*False, True, Both, Neither*	False	True	Both Entirely True AND Entirely False	Neither T NOR F
Four-Valued Logic	Epistemo-logical Tetranary	*Told False, Told True, Told Both, Untold*	Told False	Told True	Told Both Told True AND Told False	Untold Either T OR F
Normative Logic (DWDS)	Declarative Tetranary Input	*No, Yes, Both, Either*	No	Yes	Both Yes AND No	Either Yes OR No
	Declarative Tetranary Output	*Forbidden, Obligatory, Permitted, Indifferent* *or* *Not, Must, May, Should*	Forbidden or Not	Obligatory or Must	Permitted *or* May	Indifferent *or* Should
Nucleobase Logic	DNA Tetranary	*A, G, C, T*	Adenine	Guanine	Cytosine	Thymine
	RNA Tetranary	*A, G, C, U*	Adenine	Guanine	Cytosine	Uracil
Quaternary Logic	Numeric Tetranary	*0, 1, 2, 3* *or* *1, 2, 3, 4*	${}^{0}/_{3}$V *or* Fitness 1	${}^{1}/_{3}$V *or* Fitness 2	${}^{2}/_{3}$V *or* Fitness 3	${}^{3}/_{3}$V *or* Fitness 4

Although the element 'Both' (B) appears in the tetralemma and four-valued logic data models, the meaning of the term in each is very different. For example in Priest's tetralemma logic model comprising {True, False, Both, Neither}, 'Both' describes a statement that is entirely True and in the same context

27 While these summaries are my own, I am grateful to Graham Priest for brief guidance via email that helped me to generally distinguish the Aristotelian, Quantum, Fuzzy and Tetralemma logic data models. (Priest, personal communication, 2022)(Priest, personal communication, 2022) All remaining errors of interpretation are my own.

entirely False, which he names a *“dialetheia”*.[28] (Priest, 2006, p. 4). This full metaphysical superposition of the two states can be understood by considering the well-known wave-particle duality of light. In contrast Belnap et al. summarize their four-valued logic model with the set {True, False, Both, None}. In their system ‘Both’ applies to what is *said to be* entirely True and in the same context is also *said to be* entirely False. (Anderson & Belnap, 1975, 510). Their purpose is to accommodate inconsistent information from different sources arriving across a network to a computer application.

There is an additional clarification to make in regard to how ‘Four-Valued Logic’ has been expressed. I consider that Belnap et al. erred significantly in choosing the term “None” when “Untold” is their intended meaning. The detailed explanation throughout pages 490-530 of *Entailment, Vol. 2* suggests {True, False, Both, Untold} as a more accurate summary than their “{True, False, Both, None}”. Knowing that a computational application has been ‘Untold’ regarding whether a statement is ‘True’ or is ‘False’ (*The test result for this falsifiable hypothesis has not yet been reported.*) is entirely different from knowing that the statement is ‘Not True’ and is also ‘Not False’ (*The hypothesis is stated in such a way that it is not falsifiable.*). Without their explanatory text, the term ‘None’ within their set is ambiguous. To make matters worse, at one point they state incorrectly that ‘None’ “gives no information at all” (p. 512). But surely, knowing that some data is missing (*We know that the test result for this falsifiable hypothesis has not yet been reported.*) is more than not knowing whether or not the data is missing or even obtainable (*We don’t even know if this falsifiable hypothesis has been tested.*).

I suggest that the meaning of ‘Untold’ (untold True also untold False) and ‘Both’ (told True also told False) in the Belnap et al. logic model can be illustrated with a coin flip. When the coin is flipped and before it is uncovered, the outcome is 'Untold' even though it is known to be in the actual physical position of only Heads, or only Tails at that moment. However, if two credible people are permitted to observe it, one reporting 'Heads' and the other reporting 'Tails', a fourth person who has not seen it could rationally *think of* the coin as being in epistemological superposition of Both ‘ToldHeads-AND-ToldTails’, and rationally *act as if* the coin is in metaphysical superposition of Both ‘PresumedHeads-AND-PresumedTails’. In the Belnap et al. model, that fourth person is a computer.

On a separate matter requiring clarification, in Table 10 I have categorized ‘Quaternary Logic’ separately from ‘Nucleobase Logic’. Even though some of the DNA and RNA systems use four numerals, they do not represent a sequence of four quantities. For example in a short paper entitled *“Logic Core of Genetic Code”*, Zvonimir Damjanović introduced arbitrary numbering for the DNA and RNA elements

28 Priest created the term *“dialetheia”* from the Greek term **αλήθεια** (*aletheia)* means truth, honesty, transparency, and **διά** (*dia*-) meaning across.

{A,G,C,U} and {A,G,C,T} as {0,1,2,3} (Damjanović, 1998) which in subsequent years he and Miloje Rakočević developed further (Damjanović & Rakočević, 2005) (Rakočević, 2018). Similar numbering was independently assigned by Georgios Sirakoulis and colleagues, and granted, this is employed numerically. But that is just for error-checking: "Since in the double DNA strand A binds with T and C with G, by choosing the above representation the sum of bases at each base pair in the double strand is 3." (Sirakoulis et al., 2004, p. 13) (Mizas et al., 2008) Recently Qiang Liu et al. summarized ongoing developments in a 2022 article entitled "DNA-Based Molecular Computing, Storage, and Communications", in which {A, G, C, T} are assigned the binary numbers {00, 01, 10, 11}. This is not for a quantitative purpose, but only to serve as symbols to express YES, NOT, AND and OR, as explained in the following:

> "Boolean logic gates operate the basic functions YES, NOT, AND, and OR functions to perform more complex computing tasks. Thus, logic gates design plays a dominating roll in DNA-based Molecular computing, which rely on DNA SDRs [strand displacement reactions]. Different SDRs provides a variety of design for a library of logic gates. In addition, a set of more complex intelligent toolboxes are developed based on logic gates design of DNA computing. ... With the design of basic Boolean gates, including YES, AND, NOT and OR gates, more complex gates, which combines two or more basic gates, as the extension of logic gates library. ... Quaternary nucleobase coding is the most efficient approach, since each kind of nucleobase carries specific information. For an instance, the data bits "00," "01," "10," "11" could be encoded as "A," "G," "C," "T," respectively." (Liu et al., 2022, p. 901-902, 905)

Let us wrap up this section on diverse tetranary logic models with a comment on why this categorization is offered. It has long been intuitive for experienced informatics designers to ensure that their applications, databases and networks can elegantly handle incomplete, indeterminate and inconsistent data. Implementers who fail to do so find that their applications stall, or deliver incorrect or biased results, or conceal and perpetuate data problems. But it may seem irrational for systems designers to deliberately include indeterminacy and inconsistency for any reason other than 'error management'. And yet, Columbia Law School professor Susan Sturm encourages "designing for paradox":

> The paradox literature offers a third overall approach … designing for paradox. This step means building the environments and structures, and 'choice architecture' that will facilitate productive engagement with the contradictions and connections between legality and proactive lawyering as part of solving problems. (Sturm, 2019, p. 64)

In my experience paradox tends to be straightforward for most data scientists to accommodate in their design of functional systems, but uncomfortable for most software developers. Our present design employs a simple and explicit approach: a variable can be assigned for the indeterminate state Yes-OR-No, and a variable for the contradictory state 'Yes-AND-No'. These enable the expression of a rule such that:

- GIVEN the present context;
- WHEN the present set of Input Conditions is, or seems, undetermined or contradictory;
- THEN this set of Output Assertions is invoked.

4.3 Methods for Rule Logic Processing: Distinguishing the Present Design Objective

In various ways, the function that this design research undertaking sets out to create resembles the functions of existing *rules engines*, *workflow processes*, *decision-support systems*, *programmable logic controllers*, and *artificial intelligence*. But my objective differs from each of these in essential ways. It is not intended to compete with or replace any of them. Instead, the Data With Direction Specification (DWDS) is intended to fulfill a specialized function that all such systems can incorporate in-part or in-whole, and benefit from. Our contribution is intended to fulfill an auxiliary function to any of the following. This section is only intended to distinguish DWDS from some apparently similar methods and functions for the purposes at hand, and it does not imply that DWDS is 'better'.

4.3.1 Distinguishing DWDS from Programmable Logic Controller (PLC) Systems

The purpose of the present design is to bring to distributed networks a sort of general purpose logic gate. Functionally this would resemble Richard Morley's programmable logic controller (PLC) designed in the 1960s which supplied the following sequence to individual machines :

[input data]* → *[logic gate]* → *[output data] (Brown, 2015) (Amin & Mridha, 2020)

The PLC requires *known* classes of agents, scenarios and rule sequences that can be implemented in individual, cascading series and parallel multiplexed configurations. The I/O programming of PLC logic is done in an imperative style with tightly integrated data, logic and procedure.

Instead we sought to design a specification suited to the uncertainty of open self-organizing distributed networks comprised of *unlimited* agent types, partially *unknown* scenarios, and *externally defined* rule structures and sequences. For this sort of logic controller to work it is essential to unbundle data, logic and procedure, and to express the input-output (I/O) logic tables in a platform-agnostic tabular declarative form.

In Chapter 6 the DWDS logic gate is employed most generically when the upper and lower portions of a vertical I/O table are assigned the context-agnostic INPUTS and OUTPUTS meanings. There can be any number of variables in either or both the upper or lower portions of the vertical I/O table, but here we illustrate with four and four:

- INPUTS / OUTPUTS : { No | Yes | Yes-AND-No | Yes-OR-No } / { w | x | y | z }

Programmatically, the labels that are assigned to the input part of the a data package dynamically signal a particular st of outputs assigned to the bottom portion of the vertical I/O table. Operationally the DWDS describes a general-purpose multi-valued logic gate (Ebrahimi et al., 2016) for data processing on any type of computational network. Our use of this structure assigns normative terms (not, must, may, should) to the output portion of the logic gate, but this is just one of unlimited case classes. Some options are:

- CONDITIONS/ASSERTIONS: { No | Yes | Yes-AND-No | Yes-OR-No } | { not | must | may | should }
- OBSERVATIONS/EXPECTATIONS: { No | Yes | Yes-AND-No | Yes-OR-No } | { w% | x% | y% | z% } distribution quartiles
- ASSUMPTIONS/CONCLUSIONS: { No | Yes | Yes-AND-No | Yes-OR-No } | {deductive | inductive | hypothetical | transductive}

FUNCTIONAL VIEW

```
[transitory input data] ↘
      [sift1 boundary] → [sift2 classification] → [sift3 logic gate] → [output data]
[persistent input data] ↗
```

Data obtained from a transitory event is used as input to dynamically generate a sieve across a persistent reserve of rules. Sifting is first done with metadata for rules 'in effect', and second with classification data for rules that are 'applicable. Then the logic data of each row is sifted, to leave only the normative propositions which are deemed by rule authors to be 'invoked' for the original transitory input data. One could consider this to be a three-step *approximation algorithm* run on the distributed Internet that mimics orthogonal electronic filtering of integrated circuits (Padmanabhan et al., 1996) (Rao & Kailath, 1984).

These steps can be represented in terms of the semantic communication sequence summarized as: IS + RULE ⟾ OUGHT, where the vector symbol '⟾' represents all three of the sifting steps in this I/O data processing pipeline:

DATA PROCESSING PIPELINE VIEW

```
   [sieve input: IS] ↘
      [sift: in effect] → [sift: applicable] → [sift: invoked] → [output: OUGHT]
[mixed input: RULES] ↗
```

The same sequence can also be considered in terms of the GIVEN-WHEN-THEN architectural pattern from behaviour-driven development (North, 2006). This view of DWDS relates an empirical context [GIVEN 'a'] and an empirical circumstance [WHEN 'b'], with a set of declarative propositions [THEN 'c'].

BEHAVIOURAL VIEW

```
     [event] ↘
       [given] → [when] → [then] → [actions]
[assertions] ↗
```

It is also useful to consider this sequence in terms of the three implementable components of DWDS: the RuleMaker application which supplies rule resource data, RuleReserve which runs two sieves for classification, and RuleTaker which has one sieve for logic.

COMPONENT VIEW

***[substrate app]* ↘**

[RuleReserve *: boundary sieve → classification sieve]* **→**

[RuleTaker *: logic sieve]* **→ *[substrate app]***

***[RuleMaker]* ↗**

4.3.2 Distinguishing DWDS from a Rules Engine

The DWDS is similar to a rules engine in that "the system runs through all the rules, picks the ones for which the condition is 'yes', and then evaluates the corresponding actions" (Fowler, 2009). Also like a *Rule-Based Calculation Engine*, "calculations can be performed using formulae that are obtained from the database instead of being hard-wired into any program code" (Marston, 2001). However those platforms are generally presumed to operate with a central application and database of rules-as-code, all in one generic expressive language such as Python, Lisp, JavaScript, Prolog or a domain-specific language like RuleML or XBRL-Formula. For end users to provide the run-time variables, their applications must adjust to the API of the rules engine.

Our design represents a different sort of pursuit: a general-purpose specification for communicating rules as data, in a manner that that is equivalently usable by any application, in any language, on any device, without retrofits or refactoring. No particular type of 'rules engine' is required to process such data, because records conformant with the DWDS will run in any environment. A peer-to-peer decentralized RuleReserve network lets each autonomous node dynamically accommodate any of the APIs of the edge applications that are operated by end-users. Control remains at the edge.

Many rules engines select and chain together rules using statistical inference (inductive inferential logic) to arrive at conclusions with high or low confidence. (Donoho, 2017) (Holland et al., 1986) (Ormerod, 2009) (Batanero & Díaz, 2006) However the DWDS is intentionally more direct. Using no inference, it relies only upon strictly declarative conditions as input for primitive exact data-matching sieves, where particular sets of input conditions determine the output assertions. Advanced methods can be added as required by end users or intermediary service providers with auxiliary applications, but it is important that the network service itself remains uncomplicated.

4.3.3 Distinguishing DWDS from a Decision-Support System

Informatic decision systems are generally understood to be based upon “a series of if-then-else condition pairs” (Bidgoli, 2015, p. 268) in which a *probabilistic contingent* (IF) future state leads to consequent (THEN) *declarative* requirements (Mladenic et al., 2012, p. 8-9) (Keen & Scott Morton, 1978).

In contrast, a system based on the DWDS operates with data generated from an *empirical* input state (GIVEN *a,b,c* and WHEN *d,e,f*), that is used to sift consequent *declarative* output requirements (THEN *g.h.i*), in order to inform *contingent* users (North, 2006) (Fowler, 2013). The result is a simple normative assertion:

GIVEN context data;
    WHEN particular data also appears;
        THEN certain output statements are invoked.

The DWDS incorporates no ‘decisions’. It only relays normative propositions (which can including test results), so that end-users can make more informed decisions, and at their own option, automate them. Our “human-centred design” (Mitchell, 1996) orientation ensures that decisions remain with end-user human agents in the rule-maker agent and rule-taker agent roles, directly or via machines they control.

4.3.4 Distinguishing DWDS from Artificial Intelligence

Three decades ago Thomas Gruber described an "automated knowledge acquisition" method (Gruber, 1989) and he subsequently built the earliest prototype of what may be referred to as ‘artificial intelligence’. To use this phrase, it is helpful to consider the general ontology of Anthony Liew, slightly adapted (Liew, 2013):

- *data:* semantic signal acquisition
- *information:* data acquisition + contextual comprehension
- *knowledge:* information acquisition + actionable purpose
- *intelligence:* knowledge acquisition + complex inductive reasoning
- *wisdom:* intelligence acquisition + social/psychological motivation

Artificial intelligence (AI) is widely defined by machine-based knowledge acquisition and complex inductive reasoning to resolve information gaps in order to guide action. The most common reasoning methodology referred to in this context is stochastic variational inference (Hoffman, 2013)(Plötz et al., 2018).

The scope of the DWDS is only to enable the communication of normative propositions, without inference and without usurping the prerogatives of others to guide action. We refer to this as Artificial Naïvety and represent it with the slashed zero symbol as A∅ (*eh-nought*), which in mathematics denotes an empty set—a set with zero elements { }. Matters relating to context, comprehension, purpose, reason, learning or motivation are all deemed to be end-user prerogatives, to be processed at the edge of the network. In this design, operators of RuleReserve nodes provide nothing other than data structuring and storage, and run-time request-response sift-and-transmit services. Each node uses passive data from transitory messages to perform sifting operations (aka 'signal matching') upon persistent rules in storage. The signal matching advantages will be discussed further in section 9.1: *Computing Fast and Slow*; here we will focus on the empty set ∅ design requirement.

A RuleReserve node has no methods to retain or copy any data or metadata concerned with context, comprehension, purpose, reason, learning or motivation. All it does is return a response to the source of the run-time request. The requester is responsible for their own auditable activity logs.

An Internet router may be considered an example of general-purpose A∅ service. A well known axiom of Internet architecture is that: "The Internet has smart edges, computers with operating systems, applications, etc., and a simple core, which consists of a control plane and packet forwarding engines." (Bush & Meyer, 2002) David Clark, who chaired the original Internet Architecture Board (1981-1990) has referred to "architectural minimality—that is, to specify as little as possible for subsequent mechanisms to meet the goals of the architecture". (Clark, 2018, p. 67) He has explained that "the intermediate packet switching nodes, or gateways, must not have any essential state information about on-going connections." (Clark, 1988, p. 108) The Internet ideal[29] shifts any but the most basic signal processing away from the Internet protocol suite core, and leaves knowledge-oriented computational control to users and their devices at the edge. (Carpenter, 1996)

The DWDS holds A∅ to be a normative mandatory system criterion whereby its designers, suppliers and operators MUST maintain 'arms length' separation from the processing of any but the most basic data sorting and data transformation processes, because 100% of the system's data, metadata and computational control prerogative belong to end-users.

29 "Net Neutrality" (Wu, 2003)(Wu, 2003)(Wu, 2003)(Misra, 2015)(Misra, 2015)(Misra, 2015) is compromised to the extent that router designers, suppliers or operators deviate from the premises of A∅ as this is defined here.

4.3.5 Distinguishing DWDS from Business Process Workflow

A business process workflow is an interactive step-wise sequence of tasks (Gantt, 1919) (Geraldi, 2012) in which complicated, voluminous or simple consistent tasks may be automated with various types of hardware and/or software, while decisions are performed along the way by people. Two comprehensive methods for describing workflow are Business Process Modelling Notation (BPMN) and Process Chain Network (PCN) (Kazemzadeh et al., 2015). Both of these describe responsibilities and communications among participant entities and subsystems, but the orientation of BPMN is to communicate workflow structure (Enstrom, 2016), whereas PCN is for workflow performance improvement (Sampson, 2015).

The DWDS keeps the 'rule-maker agent' authoring the rule input at functional 'arms length' from the 'rule-taker agent' using the output directly or through their machine. Unlike the methods discussed in Section 4.1 specifically for expressing and processing rule logic, a "data processing pipeline" is a generic pattern involving an uninterrupted flow through a single computational input-output sequence (von Landesberger et al., 2017). Although intermediate output states of the data become inputs to subsequent steps, this pattern involves no sequence of decisions. DWDS's RuleReserve network and decentralized RuleTaker applications each implement this simple, scalable, fault-tolerant design pattern which automatically sifts and sorts large volumes of distributed data *in a single pass* (i.e. without 'loop' or 'if' statements) using parallel processing across as many available platforms as required. The decentralized sifting process of DWDS is comparable to, but functionally much simpler than the well-known map/reduce data pipeline across a data fabric (Zeng & Plale, 2013) (Dean & Ghemawat, 2008b) (Cao et al., 2010) (Swedlow et al., 2011) (Li et al., 2016) (Maitrey & Jha, 2015) (Dong et al., 2018).

4.3.6 Distinguishing DWDS's RuleData Model from a Domain Specific Language

The DWDS introduces a new structured way for rules to be expressed, communicated and operationalized using any natural language. RuleData is less than a *'domain-specific language'* (DSL).[30] It is merely a syntactic specification to support a particular style of tuple-oriented programming (Underwood, 2011). In our reference implementations this data is packaged with JSON syntax (JavaScript Object Notation). (Bray, 2014) The same data can be equivalently expressed in another implementation using CBOR (Concise Binary Object Representation, which is loosely based on JSON) (Bormann & Hoffman,

30 A *general-purpose language* has the syntactic flexibility and semantic range to enable the transmission of any domain of information. A *domain-specific language* may have a syntactic structure tailored to particular communication functions, or be bound to a semantic schema for well-circumscribed categories. This is not a strict distinction, since as Marjan Mernik et. al. explain, "domain-specificity is a matter of degree". (Mernik et al., 2005, p. 5)(Mernik et al., 2005, p. 5)(Mernik et al., 2005, p. 5). *'Stand-alone'* domain-specific languages are developed from scratch when the purpose is to facilitate creation, maintenance and control of methods by subject matter specialists who are normally not programmers. These may be "executable in various ways and to various degrees, even to the point of being non-executable" (Mernik et al., 2005, p. 6) (Mernik et al., 2005, p. 6) (Mernik et al., 2005, p. 6) (Mernik et al., 2005, p. 6) . *'Integrated'* domain-specific languages are structured extensions of existing general-purpose programming languages, in order to expand implementation flexibility for programmers while still working within the same technology platform (Havelund et al., 2010).

2013), or CSV (comma separated value), or as separate database fields without any associated syntax.

By separating each rule sentence into a reusable set of six syntactic elements, and also separating these from an array containing the relations among conditions and assertions, our method accepts Robert Kowalski's recommended separation of logic and control:

> "Although the trend in databases is towards the separation of logic and control, programming languages today do not distinguish between them. The programmer specifies both logic and control in a single language while the execution mechanism exercises only the most rudimentary problem-solving capabilities. Computer programs will be more often correct, more easily improved, and more readily adapted to new problems when programming languages separate logic and control, and when execution mechanisms provide more powerful problem-solving facilities of the kind provided by intelligent theorem-proving systems." (Kowalski, 1979a, p. 435)

RuleData imposes no programming requirements beyond the six-element syntactic structure. DWDS imposes *no need to choose a particular programming language*. Moreover, DWDS also enables rule authors to write and read rules in their preferred vernacular, with or without domain-specific terms or style. The choice of expression style is a matter of end-user judgment, which Nathan Schneider explains:

> "We come, then, to the main question: When is it worth designing a new annotation scheme? My answer is, *When annotating with an existing scheme would be more painful (costly) than starting afresh.* The second question, What level of granularity?, is similarly answered by weighing these tradeoffs: too coarse, and the annotations will not be very informative or useful; too fine, and training annotators will be costly, the annotation will be slow, annotator reliability will be low, and some categories may be highly sparse. Estimating these trade-offs in a particular setting is a qualitative judgment call..." (Schneider, 2015, pp. 152–153)

The DWDS does, however, create a background incentive to use common schemas and lexicons, an approach which sidesteps the trend towards redundancy and inconsistency that has emerged among competing standard XML schemas (Sliwa & King, 2000). We have designed a practical incentive for semantic alignment to emerge through co-opetition (Brandenburger & Nalebuff, 1997), but that is left to emerge on its own, independently of the specification *per se*. The incentive is sufficient.

There is great value in the various domain-specific XML schemas that have been painstakingly structured and negotiated. But XML notation is optimized for the semantic Web where a browser has a small job to do in attaching semantics to displayed content. It is not optimal for high-volume, high-performance data processing. Even the 50-year-old NETL (NETwork Language) representation designed by Scott Fahlman to supply declarative real-world semantic knowledge in response to queries, would outperform XML by far in a distributed database (Fahlman, 1977) (Holland et al., 1986, p. 19). Fahlman's original explanation is worth citing at length here, because the DWDS embodies a similar way of thinking:

> “We forget about trying to avoid or minimize the deductive search, and simply do it, employing a rather extreme form of parallelism to get the job done quickly. By ‘quickly’ I mean that the search for most implicit properties and facts in this system will take only a few machine-cycles, and that the time required is essentially constant, regardless of how large the knowledge base might become. The representation of knowledge in this system is entirely declarative: the system's search procedures are very simple and they do not change as new knowledge is added. Of course, the knowledge base must contain descriptions of procedures for use by other parts of the system, including those parts that perform the more complex deductions, but this knowledge is not used by the knowledge base itself as it hunts for information and performs the simple deductions for which it is responsible. The parallelism is to be achieved by storing the knowledge in a semantic network built from very simple hardware devices: node units, representing the concepts and entities in the knowledge-base, and link units, representing statements of the relationships between various nodes. (Actually, the more complex statements are represented by structures built from several nodes and links, but. that need not concern us here.) ... The controller is not only able to specify, at every step of the propagation, exactly which types of links are to pass which markers in which directions; It is also able to use the presence of one type of marker at a link to enable or inhibit the passage of other markers. It is the precision of such a system that gives it its power, but only if we can learn to use it properly.” (Fahlman, 1977, p. 11)

The declarative non-canonical approach employed in RuleData arises from the need to process large sets of unstructured user-generated data, and this is similar to the requirements of search engines (Dean & Ghemawat, 2008b), and to the processing of natural language text (Plank, 2016). This is accomplished by constraining rule expression to a small set of metadata, and to a single syntactic structure for sentences that provide meaning to the logical relations within each rule, DWDS achieves operational simplicity.

RuleData is thus put forward as a generalized means of expressing each condition and assertion that occurs in legislation, policies, standards or agreements in a human-readable but also informatically-processable form. This can be embedded or automatically transcribed into any other programming language, making it platform-independent. Normative data expressed in RuleData must not replace or be inserted into legal documents, rather it belongs in a ‘schedule’ or some other type of attachment to a legal text. This loses nothing operationally, yet it remains subordinate to the natural language text endorsed by legislators or parties to the agreements. This way, when there is a bug to fix, it is not necessary to go back to the legislature or to the parties for re-negotiation – the original natural language text remains the legal reference.

The DWDS data processing method is described in Chapters 5 and 6. First the RuleReserve network sifts for rules ‘in effect for a context and ‘applicable’ to categories of activities and things. Then a RuleTaker component sift out rule assertions that are ‘invoked’ by particular circumstances.

4.4 Influences and Inspirations from 70 Years of Methods in Programmable Logic

This section provides a conspectus of various technical concepts and methods which in various ways have shaped the path of this research. My own design decisions were informed by analysis of these sources of ‘primary data’ about the design of programmable logic (e.g. about styles of rule expression; data processing methods). This is a general acknowledgment of available methods that I have explicitly reflected upon, and have used directly or indirectly. It includes some references to methods that helped me to think through *what I wanted to do differently* (i.e what I did not want to do).

Rediscovering and rehabilitating long-forgotten methods from the 1950s-60s-70s-80s is an important part of *this* design research undertaking. Most of the entries included here are well known, but many required meticulous tracking down of original, generally forgotten sources in an effort to ensure proper acknowledgment. In particular, tabular declarative programming required a determined excavation effort.

The original sources are arranged according to themes that I consider to be contextually relevant to my own research perspective. These cannot be assumed to emanate from how the authors would have grouped their work. This section is not intended to be a systematic or general history. It only reflects my recent journey of reading and reflection on these themes while arriving at a partially novel design.

4.4.1 Data Structuring and Transmission

1950s

- Claude Shannon’s 1948 “Mathematical Theory of Communication” (Shannon, 1948) led to information theory and algorithmic logic transitioning beyond the realm of technical methods to become acknowledged as a domain of formal design science.
- Noam Chomsky re-framed linguistics as a branch of mathematics by formalizing a generative context-free phrase-structure grammar framework in his PhD thesis *Transformational Analysis* (Chomsky, 1955), a 918-page document *The Logical Structure of Linguistic Theory (LSLT) (Chomsky, 1956)* and the short book *Syntactic Structures* (Chomsky, 1957).[31]

1960s

- John Tukey presented the case that "data analysis is intrinsically an empirical science"; because in addition to technological and statistics methods, data is central to discovering the nature of the world. His seminal work launched “data science”. (Tukey, 1962)
- Adrian McDonough published a data science strategy wherein the analyst should aim to minimize the amount of information needed, while maximizing the decision-making power it bestows. (McDonough, 1963)

31 The use of Chomsky’s 1957 context-free phrase structure grammar in our present design is explain in Section *5.3.3.2 Making the Syntactic Elements Explicit.*

1970s

- At the beginning of the 1970s the telex (teleprinter exchange) wireless radio-satellite-microwave network was in use globally by banks to route money messages using a common syntax and semantics across linguistic and systems boundaries.[32]
- Stafford Beer, Fernando Flores and Raúl Espejo led Project Cybersyn in Chile to create an informatics network of real-time dynamic supply chain data (Cyberstride) across industry sites in manufacturing, primary production, transport, storage, etc., based upon the initial 1972 version of Beer’s Viable System Model (Beer, 1981) (Andrew, 2012) (Leonard, 2015) (Medina, 2015) Telex machines would transmit live data to a mainframe computer (first an IBM System/360 then a Burroughs 3500) to provide dynamic updates to a whole-economy simulation model programmed in DYNAMO (DYNAmic MOdels) (Forrester, 1961), in order to return live feedback for supply chain optimization.

  > “The system Beer proposed worked in the following way. Intervenors would use the telex machines at their enterprises to send production data to the telex machine located at the National Computer Corporation. Chilean computer experts would then punch the data onto cards and feed them to the mainframe. The computer ran statistical software programs that compared the new data with those collected previously, searching for significant variations. If the program encountered such a variation, it alerted the computer operators, who would send the data over the telex network to CORFO and the intervenors affected. As a result, CORFO [Corporación de Fomento de la Producción / Corporation to Foment Production] would communicate with the intervenors in order to better understand the situation and help resolve the problem, if one existed.” (Medina, 2011, p. 72)

1980s

- Kark-Erik Erikkson, Kristian Lindgran and Bengt Mansson developed general-purpose quantitative measures for data science (i.e. information structure, texture and complexity, including algorithmic information) as a basis for interdisciplinary communication. (Eriksson et al., 1987) The same year the “Committee on NASA Information Systems” chaired by Adrian McDonough, published a report that set out priorities for the domain of data science research and development. (McDonough, 1987)
- A comparative review of algorithmic filtering methods with over a hundred rules was published in 1987 by James Woolley and Nicholas Stone (Woolley & Stone, 1987).

1990s

- Jon Bosak, Tim Bray, James Clark and others to created a working subset of SGML called eXtensible Markup Language (XML) in order to embed semantic schemas into Web documents, and to enable data interchange among networked heterogeneous systems. (Bosak, 1996) This technique was helpful to both the procedural imperative and tabular declarative styles of programming, and it led to widespread collaboration towards a “Semantic Web” (Berners-Lee, Hendler, & Lassila, 2001). A further step was taken by Michael Genesereth and Richard Fikes who designed the Knowledge Interchange Format (KIF) to embed first-order logic directly into Web documents (Genesereth & Fikes, 1992)

32 Each telex machine had an identity on the network so that users could direct messages using the machine’s keyboard, which punched holes in a paper tape. Each line of eight holes across a one-inch tape represented one byte of binary data (0000 0000 to 1111 1111), enabling 255 unique hexadecimal values. A ninth line of holes was for a sprocket to move the tape through the machine so that it could read the tape and convert the data to electromagnetic wireless signals. Automatic confirmation of each message was accomplished by placing a WRU (Who aRe yoU) code at the beginning of each transmission, which would automatically prompt the recipient telex machine to send an "answerback" identification message.

Early 2000s

- "Data science" came to be widely acknowledged as a distinct academic and commercial research domain beyond computational algorithms, to include information theory, scenario modelling, data systems design, and processing performance. (Cleveland, 2014) (Donoho, 2017)
- The phrase "Big Data" came to refer to the "explosive growth in data volume, velocity and variety" throughout business, academic, scientific and government domains, and new types of systems and methods to obtain meaning from it. (Diebold, 2012) The capability to efficiently process large volumes of data was enabled by advances in decision table concepts, methods and platforms of the previous 20 years. (Cunneyworth, 1994) (Janicki et al., 1997) (Garcia et al., 2000) (See section 4.4.2 Tabular Logic Programming, below.)
- The extensibility of XML ensured its adaptability to any context and any level of granularity. At first this seemed optimal, but through this decade XML schema proliferation resulted in a complicated labyrinth of competing standards. Ironically, this undermined computational simplicity and speed. Domain-specific XML schemas seem suitable when a browser has a limited job to do in attaching semantics to displayed content of an individual site. But the large number and diversity of XML schemas which had come to be designed and implemented 'bottom-up' by diverse communities led to redundancy and inconsistency for the Semantic Web as a whole (Sliwa & King, 2000).
- The need for interoperability among XML schemas led to the Resource Description Framework (RDF) which supplied a cross-platform structure for subject-verb-object triples (Brickley & Guha, 2000) (Bray, 2001). This provided the basis for Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) offering developers a way to create "sets of inference rules that they can use to conduct automated reasoning", with "a language that expresses both data and rules for reasoning" (Berners-Lee et al., 2001). This expanded as the "Policy-Aware Web" (Weitzner et al., 2006), including a rules language for the Semantic Web (N3) (Berners-Lee & Connolly, 2008).
- Several competing standards emerged for rules expression across business rule domains, languages, software systems:
    - OASIS published RuleML semantic schema standard, led by Harold Boley, Adrian Paschke (Boley, 2006).
    - The Object Management Group (OMG) published *Semantics of Business Vocabulary and Business Rules (SBVR)*, led by Ron Ross (OMG, 2005). This includes *RuleSpeak*, a comprehensive business rule notation and grammar in "structured English" (OMG, 2016b). Also James Taylor, Neil Raden, Barbara von Halle, Larry Goldberg, Ron Ross produced *Decision Model and Notation (DMN)* through the OMG to express logic rules for Business Process Model and Notation (BPMN) (OMG, 2013).
    - The ISO has published and maintained several relevant standards. ISO TC 37 *Terminology and Language & other Content Resources* structures communication using special purpose controlled natural languages. Formal logic expression was added through ISO 1087-1:2000 *Terminology work — Vocabulary*. Additional standards relating to rules expression are: ISO/TC 37/SC 4 *Language resource management*; and, ISO/TR 9007:1987 *Information processing systems — Concepts and terminology for the conceptual schema and the information base*. (ISO, 1947) (ISO, 2001) (ISO, 2000) (ISO, 1987)
    - A network of legal specialists led by Alexander Boer, Radboud Winkels and Fabio Vitali designed MetaLex as a proposed standard for jurisdiction-neutral, language-neutral XML encoding of legislation, as well as the Legal Knowledge Interchange Format (LKIF) ontology, application programmer interface, and inference engine specification for legal decision support systems and data interchange.

- ◦ Meanwhile, diverse industry sectors involved in practical rules automation, from taxation services to autonomous vehicles, pursued domain-specific standardization. The majority of this has involved procedural imperative rules-as-code libraries for particular application environments, while the tabular declarative rules-as-data style was maintained in several specialized domains.

2010s

- The phrase *"Data-Driven Transformation"* came into use in reference to normative macro-level decision-making based upon large-scale accumulation, automated processing and analysis of data structures, functions, feedbacks and boundaries. The related phrase "*data-driven development*" (Dubois et al., 2000) (Anderson, 2015) (Hoffman, 2015) (World Bank, 2019) also incorporated deontic analysis and interpretation. Global regional and local competitive markets came to be shaped by competitive control over automated data accumulation and flow. (Baker et al., 2005) (Orlowski, 2020)
- Tatiana von Landesberger, Dieter Fellner and Roy Ruddle formalized the concept of "data processing pipelines—the set of data processing steps, parameters and algorithms that perform operations on the data" as an input-output sequence. (von Landesberger et al., 2017)
- The proliferation of computerized networked devices and decision management systems led naturally to an expansion of rules automation research and deployment, requiring the communication of rules and requirements across different platforms. A commonly declared target was a single open, generic, non-proprietary meta-model for rules defined by an XML schema. But competitive systems had already been implemented and deployed, therefore standardization of metadata was impractical. Brian Stucky has commented:

> "The search for standards began with business rules and we saw approaches like RuleML, XBRL, RuleSpeak, SVBR, and many others. All great work, but none provided THE answer. With every BRMS vendor using its own proprietary representation for rule execution, and their own syntax for expressing rules across various metaphors, this task became increasingly difficult. Consequently, virtually every business rule implementation became unique – not only from company to company but often within the same company. More recent efforts, including RIF (Rule Interchange Format), can potentially narrow the gap – but the gap it still exists. ... And now we see yet another emerging standard for the new world of decision management – the Decision Management Notation (DMN). ... The stated goals for DMN are twofold: 1) to provide a notation understandable by all business users and technical developers, and 2) to ensure decision models are interchangeable across organizations via XML." (Stucky, 2020)

The drive for an interchange standard for rules data has been only partially successful. Currently there are two general-purpose rule interchange standards which are similar. They can be can be approximately auto-transcribed into each other, although they are not fully inter-operable (IBM, 2014).

- ◦ The Rule Interchange Format (RIF) (Kifer & Boley, 2013) became a World Wide Web Consortium (W3C) recommendation for decision rules in mid-June 2010.
- ◦ The Requirements Interchange Format (ReqIF) originally from the automobile design industry, expanded into other engineering and manufacturing fields, and in early 2011 became a specification of the Object Management Group (OMG, 2016a).

- Formal logic expression in XML arose from two philosophically different directions:
  - ◦ A *general-to-specific* conceptual trajectory of RuleML arose from primary research led by Harold Boley to create a general purpose method for network communication of rules in a manner that can be modelled, automatically validated and functionally operated (Boley, 2006) (Boley et al., 2010) (Boley et al., 2017). This has been extended to diverse domains.

    - A *specific-to-general* trajectory of the RASE Method (Requirements-Applicabilities-Selection-Exception) originated with applied architectural engineering work by Eilif Hjelseth and Nick Nisbet to automate compliance validation of algorithmic building information models (BIM) (Hjelseth & Nisbet, 2010) (Hjelseth & Nisbet, 2011). This has since been extended to additional domains (Schartum, 2016a).

- The Object Management Group adopted RuleSpeak guidelines for business rule expression in Structured English, as a core part of its standard on Semantics of Business Vocabulary and Business Rules (SBVR) (OMG, 2005) (OMG, 2019b) (Annex A - SBVR Structured English; Annex B - SBVR Structured English Patterns; and Annex H - The RuleSpeak Business Rule Notation (OMG, 2016b)

4.4.2 Tabular Logic Programming

1970s

- Jonas Rabin organized through CODASYL to establish in 1973 the “Decision Table Task Group”, a multi-stakeholder forum to assemble and refine the concepts and methods of expressing Kowalski’s procedural logic in tabular form, to generally improve their usability, and to develop a strategy for promoting their use.
- In 1975 Dick Morley's team at Modicon introduced the first programmable logic controller (PLC) driven by a microprocessor.
- Art Lew and Doris Tamanaha demonstrated that “any Turing Machine program can be ‘emulated’ by Minsky’s *procedural* decision tables. This is done by letting each Turing Machine instruction of the form (input,state)+(output, tape movement, state) be represented by a decision table rule where (input,state) are conditions and (output,tape movement,state) are actions.” (Lew & Tamanaha, 1976)
- Scott Fahlman designed NETL (NETwork Language) knowledge representation system which stored real-world, common-sense knowledge in a massively parallel network of simple elements (Fahlman, 1977), essentially to “supply declarative information in response to queries.” (Holland, 1999)

1980s

- A comprehensive 270-page report *“A Modern Appraisal of Decision Tables”* was published by CODASYL’s “Decision Table Task Group”, to advance tabular declarative input/output programming methods for general-purpose computing (Jorgensen & Marselos, 1982).
- A concise view of the approach from this period is provided in the work of Art Lew (Lew, 1982) (Lew, 1983).

1990s

- Fred Brooks explained at the time: “Show me your flowcharts and conceal your tables, and I shall continue to be mystified. Show me your tables, and I won't usually need your flowcharts; they'll be obvious.” (Brooks, 1995)
- In 1992 Jan Vanthienen and Elke Dries published *Developments in Decision Tables: Evolution, Applications and a Proposed Standard* with a generic and a formal definition, in the procedural declarative style (Vanthienen & Dries, 1992):

> “A decision table is a tabular representation used to describe and analyze procedural decision situations, where the state of a number of conditions determines the execution of a set of actions. ... all distinct situations are shown as columns in a table, such that every possible case is included in one and only one column (completeness and exclusivity).”
>
> “The decision table is a function from the Cartesian product of the condition states $CR = CT_1 \times ... \times CT_{enum}$ to the Cartesian product of the action values $AR = AV_1 \times ... \times AV_{anum}$, by which every condition combination $x \in CR$ is mapped into one (completeness) and only one (exclusivity) action configuration $z \in AR$.”

- One of the most prominent industry implementations of tabular declarative programming was on mainframes for global banking, financial services and industrial organizations, designed by Wayne Cunneyworth and William Olders. They combined the methods of Datalog with input/output tables to express algorithm logic gates, which could be stored pre-parsed, ready for execution by an interpreter. Variables were completed during execution with incoming transaction data and in-memory look-up tables. These tables could be fixed or variable length, and were also portable across different interpreters. The method is explained in “Table Driven Design: A development strategy for minimal maintenance information systems”. (Cunneyworth, 1994)
- Concurrent work in tabular declarative programming was led by David Parnas, Jan Madey and Michal Iglewski. They were concerned with the documentation of nuclear energy management systems, to facilitate pragmatic program auditing and overall ease of maintenance. (Parnas et al., 1994)
- Ron Ross published a specification for ‘RuleSpeak’, a notation for rules based on structured English with a set of re-usable semantic and syntactic patterns. (Ross, 1997)
- John Chambers and Rick Becker led development of the S language for tabular “programming with data” which was structured as subject-predicate-object triples (Chambers, 1998). This became more widely used in geomatics, health sciences and econometrics through the free/libre/open derivative R language led by Ross Ihaka and Robert Gentleman (Ihaka & Gentleman, 1996). Today R remains a widely used platform for data analytics. In a 1993 article Chambers distinguished fundamental *learning from data* (“Greater Statistics”) from superficial *statistical methods* (“Lesser Statistics”). He explained: “It is our version of the scientific method, with an emphasis on empirical understanding and a healthy skepticism about theory for its own sake: in short, a methodology for learning from data.” (Chambers, 1993)

Early 2000s

- Leo Breiman described two cultures in data science: *data modelling* (goodness of fit) in academic venues, versus *algorithmic modelling* (decision trees/tables) in business and industry.

> “The approach is that nature produces data in a black box whose insides are complex, mysterious, and, at least, partly unknowable. What is observed is a set of x’s that go in and a subsequent set of y’s that come out. The problem is to find an algorithm f(x) such that for future x in a test set, f(x) will be a good predictor of y." (Breiman, 2001)

- Ana Moreno Garcia and Jan Vanthienen provided an annotated bibliography on the decision table literature from 1982 through 2000. (Garcia et al., 2000) A comprehensive *"History of Modelling Decisions using Tables"* is maintained online by Jan Vanthienen (Vanthienen, 2012a) (Vanthienen, 2012b) (Vanthienen, 2012c).
- Wes McKinney added Pandas ('panel data') tabular programming capabilities to the otherwise procedural Python programming environment.

2010s

- Reynold Xin designed 'data-frames' to bring tabular declarative programming capability to the Spark data processing platform.
- Avinash Lakshman and Prashant Malik designed the Cassandra Query Language (CQL) 'wide-column' database format to optimize the layout of *ad hoc data for predetermined query types* (Carpenter & Hewitt, 2016). This is unlike the Standard Query Language (SQL) 'relational' data format that is optimized for *ad hoc queries against a predetermined data model*. Relational databases place related fields into named tables with a unique identifier distinguishing each record. In contrast, wide-column databases place related facts into columns of flexible tables ('data fabrics') and employ the column names as the search keys. In the wide-column data store there are as many columns as there are data types, and each row contains a unique record that may have data or null in each column.

4.4.3 Procedural Logic Programming

1930s

- Alan Turing described a "universal computing machine". (Turing, 1937)

1950s

- Alfred Horn designed a novel method for 'logic programming' to resolve first-order logic with *declarative* subject-verb-object triples: a subject via a predicate documents or modifies the properties of an object. (Horn, 1951) (The Latin term *praedicatum* means "that which is said of" something.) Horn's method is based on 'proof-by-contradiction': expressing all assertions as false, then deriving contradictions in order to deduce which ones are not false. This computational strategy enables very fast computer algorithms to test for logical consequence.
- The first programming language to systematically implement Chomsky's context-free grammar structures was ALGOL (ALGOrithmic Language). John Backus was the first to separate syntactic from semantic structure, which led to 'Backus Normal Form' (BNF), refined with contributions from Peter Naur, as the core of all subsequent procedural programming languages for computers (such as C and its derivatives C++ and C#, as well as PHP). ALGOL60 was designed through a multilateral trans-Atlantic collaboration involving dozens of contributors jointly convened by the Association of Computing Machinery (ACM) and UNESCO. (Backus et al., 1960)
- Through CODASYL (Committee on Data Systems Languages) Jean Sammet, the lead designer of FORMAC (FORmula MAnipulation Compiler) and Grace Hopper, the lead designer of FLOW-MATIC, convened a team that jointly designed the first general-purpose platform-agnostic programming language called COmmon Business-Oriented Language (COBOL). Its formal syntactic notation and formal semantic definition techniques were guided by Noam Chomsky's linguistic concepts and techniques, in particular the idea that a small set of simple, active, declarative sentences could form a semantic kernel from which all other sentences could be constructed by various transformations. Hopper explained later in an interview: "That was the beginning of COBOL, a computer language for data processors. I could say 'Subtract income tax from pay' instead of trying to write that in

octal code or using all kinds of symbols.” (Hopper, 1981, p. 3) This functioned with a table of logic expressions so that: "The programmer may return to being a mathematician ... supplied with a catalogue of subroutines. ... Make a list of arguments and results and number them. ... The order is immaterial, so that forgotten quantities can be added at the end.” The programmer “does not even need to know the particular instruction code used by the computer” as it is sufficient “to be able to use the catalogue to supply information to the computer” about the problem. (Hopper, 1987, pp. 273–275)

- John McCarthy designed LISP (LISt Processor) based on linked lists so that a standard data structure could be used to represent the structure of a program.

1960s

- Marvin Minsky designed decision tables to illustrate “the finite-state parts of our [Turing] machines with a *procedural* system of quintuples “(old state, symbol scanned, new state, symbol written, direction of motion)” involving the elements {0,1,B,H,-} where ‘B’ ≡ break, ‘H’ ≡ halt, and ‘-’ ≡ null. He used decision tables to illustrate how “the finite-state parts of our [Turing] machines can be described nicely by sets of quintuples of the form (old state, symbol scanned, new state, symbol written, direction of motion) ... i.e. (q_i, s_j, q_{ij}, s_{ij}, d_{ij}) ... i.e., as quintuples in which the third, fourth, and fifth symbols are determined by the first and second” (Minsky, 1967, p. 119).
- Richard (Dick) Morley designed the first programmable logic controller (PLC) in 1964 for repeatable functions such as timing, counting, calculating, comparing, and signal processing. PLCs were designed to replace hard-wired relay control systems to enable automation with programmable [input] → [logic gate] → [output] relations. The basic logic gates AND, NOT, OR and XOR (exclusive OR allowing either but not both) could be used individually or in combination, to receive input parameters from various sensors, keyboards or switches, and given the structure of the logic gate, result in the determined output parameters in order to control motors, valves, switches or subsequent PLCs. Unlimited combinations of serial and parallel PLCs could thus enable flexible automation, which revolutionized industrial and manufacturing automation. Production began in 1969 under the name Modicon (modular digital control).
- John Alan Robinson published “A Machine-Oriented Logic Based on the Resolution Principle” describing the application of Horn’s strategy in applied computing. (Robinson, 1965)
- Charles Goldfarb led an effort to enable machine-readable documents across government, law, and industry with a specification for embedding *declarative* tags into natural language texts and other forms of data. This would enable the automated generation of logic statements from well-structured natural language texts, although it would require an uncommon level of discipline among the natural language authors.

> “We were trying to do an automated law-office application. I had been a lawyer (in fact, I still am). Lawyers must do research on existing case law, decisions of court, and so on, to find out which ones are applicable to a given situation, find out what the previous legal rulings have been, and then merge that with text that the lawyer has written himself. Eventually, if it's, say, a brief for the court, [he must] then compose it and print it. At the time, which was 1969 or 1970, there weren't any systems available that did these three things. So in order to get the systems to share the data we had to come up with a way to represent it that was independent of any of those applications.” (Floyd, 2009)

1970s

- In 1974 Robert Kowalski published the short article *Predicate Logic as Programming Language* in which he first described a *procedural logic* interpretation of Alfred Horn’s *deductive logic* clauses. This could be used to relate contingent input states to programmable instructions to computers. (Kowalski, 1974) Lloyd explains the significance:

  > “The idea that first order logic, or at least substantial subsets of it, could be used as a programming language was revolutionary, because, until 1972, logic had only ever been used as a specification or declarative language in computer science. However, what [Kowalski] shows is that logic has a procedural interpretation, which makes it very effective as a programming language.” (Lloyd, 1987, p. 2)

  Then in 1979, Kowalski published the 250-page *Logic for Problem Solving*, which provided a foundation for programming languages, database design and intelligent systems.

- Alain Colmerauer and Philippe Roussel expressed Kowalski’s procedural logic in the language Prolog (PROgrammation en LOGique).

1980s

- Marek Sergot, Robert Kowalski et.al. programmed the British Nationality Act in Prolog as an experiment to test its suitability for expressing legislation. As part of this first significant experiment in “rules-as-code”, the team also programmed various sections of legislation on immigration, taxes, subsidies, pensions and employee compensation. The main limitations they reported had to do with interpreting negation as failure; the difficulty handling counterfactual conditionals (a state that is not, but could or would have been); and various complex and commonsense issues of knowledge representation needed to understand the legislation. (Sergot et al., 1986)

- Hervé Gallaire, Jack Minker, David Maier and David Warren collaborated to create Datalog as a simpler subset of Prolog, consisting solely of declarative facts and rules, without operational functions. Rules are expressed as two-part clauses: the facts about an event or state and its logical implications. Facts alone are expressed with no implications. (Lloyd, 1987)
  - A fact asserts with a *predicate* that a stated *object* is true for a particular combination of stated *subject* values.
  - A rule asserts that whenever certain facts occur, then at least one additional fact is asserted in a derived relation. (Maier et al., 2018, pp. 3–5)

  This provides the basis for programming with first-order logic. Datalog separates statements of logical relations from resolution procedures, so that programmers can focus on specifying the logical relations with purely declarative facts and rules. The machine optimizes how each problem is to be solved using its available procedures. John Alan Robinson described the duality of deductive and procedural logic:

  > “I know that it's literally true that a function is just a special kind of relation. But you can turn that around, and you can observe also with equal merit that a relation is just a special kind of function. As a matter of fact, that's how [Gottlob] Frege saw it. For him, a relation is a function from tuples of things to truth values. And so, you think of evaluating a relation in just the same way as you think of evaluating any other function. It's just a different target domain.” (Robinson, 1983, p. 114) (Frege, 1879, §9)

- Wider interest in Goldfarb’s framework for embedding tags in rules of government, law, and industry led to its status as the global Standard Generalized Markup Language (SGML) (ISO, 1986).

1990s

- The divergence in methods for predicate logic that emerged in the 1970s changed character in the 1990s. The market for procedural programming tended towards a style of imperative instructions, whereas declarative programming tended towards a tabular style.
    - *Procedural imperative programming* found its niche in general market computing where programmers, as 'software engineers', are expected to rapidly show 'good-enough' prototypes for highly competitive milestone-driven clients. Step-wise developer productiveness is measured by speed of delivery and the number of process functions built into a system.
    - *Tabular declarative programming* found its niche in large-scale, complex industry use cases in which operational integrity is mission-critical (banking, nuclear control systems, insurance). In such scenarios, analysts and programmers, as applied 'logicians', are provided the time to understand and solve for whole system problems. Tabular expression of both the metadata and logic simplifies documentation, support, auditing and maintenance.

Early 2000s

- A concise 20-year retrospective editorial comment by Moshe Vardi, Editor-in-chief of *Communications of the ACM* explored the roots of the distinction between *deductive* versus *procedural* programming. His assessment *"What is an Algorithm?"* (Vardi, 2012) reviewed two seminal articles with the same title:
    - A paper by Yiannis Moschavakis entitled *"What is an Algorithm?"* defines it as any procedural expression designed to accept input data *x*, and to return output *f(x)* as further input. It is therefore referred to as a 'recursor' which terminates once a specified condition is reached (Moschovakis, 2001). This view underlies the *imperative procedural* programming style.
    - A paper by Yuri Gurevich also entitled *"What is an Algorithm?"* considered this to refer to any instruction for storing data in a particular state. From input data *x*, this 'abstract state machine' generates a resulting output *f(x)* (Gurevich, 2014). This view underlies the *tabular declarative* programming style.
    - Vardi accepts both of these with the following rationale:[33]

> "So is an algorithm an abstract state machine or a recursor? Mathematically, one can show that recursors can model abstract state machines and abstract state machines can model recursors, but which definition is primary? … An algorithm is both an abstract state machine and a recursor, and neither viewed by itself fully describes what an algorithm is. This algorithmic duality seems to be a fundamental principle of computer science." (Vardi, 2012, p. 5)

33 Both interpretations are consistent with Robert Kowalski's 1979 paper "Algorithm = Logic + Control"

- Ian Grigg published *The Ricardian Contract*, a method to identify and describe the issuance of financial instruments as contracts. This arose from earlier design work by Grigg and Gary Howland on the Ricardo payment system.

> "Whereas other issues have contracts, our issues are contracts. Our innovation is to express an issued instrument as a contract, and to link that contract into every aspect of the payment system. By this process, a document of some broad utility (readable by user and program) is drafted and digitally signed by the issuer of the instrument. This document, the Ricardian Contract, forms the basis for understanding an issue and every transaction within that issue. By extension, all issues of value, such as currencies, shares, derivatives, loyalty systems and vouchers, can benefit from this approach." (Grigg, 2004)

- Algorithmic systems began to out-perform humans in complex decision environments (Das et al., 2001). David X. Li's Gaussian copula function came to dominate the algorithmic derivatives which supplanted market pricing based on human determination of price (Li, 2000), although he was concerned about unwarranted faith in his work and he cautioned in 2005: "The most dangerous part is when people believe everything coming out of it (Whitehouse, 2005)." By 2008 three quarters of global financial liquidity was expressed as derivatives, ballooning to more than seven times global Gross Domestic Product (BIS, 2008). In the midst of the 2008 financial implosion, a short pseudonymous paper "Bitcoin: A Peer-to-Peer Electronic Cash System" (Nakamoto, 2008) introduced the algorithmic "blockchain" method of recording and verifying transactions.
- In 2009 Emilian Pascalau and Adrian Giurca introduced *JSON Rules - The JavaScript Rule Engine*, a procedural event-condition-action (ECA) capability to handle DOM (Document Object Model) events within the browser. (Pascalau & Giurca, 2009)
- Jeffrey Dean and Sanjay Ghemawat integrated data mapping and data reduction functions of John McCarthy's LISP framework into a simple, scalable, fault-tolerant 'map/reduce' data processing pattern. Map/reduce enables selecting and sorting through large volumes of distributed data in a single pass (i.e. without 'loop' or 'if' statements) using parallel processing across any number of platforms. (Dean & Ghemawat, 2008b, p 107)

4.4.4 Tetranary Computing

1930s-1960s

- Nuel Belnap completed his Ph.D. dissertation "A Formalization of Entailment" which provided a framework to modern "four-valued logic". (Belnap, 1959) His thesis printed and distributed the following year as a technical report by the Office of Naval Research.
- Timothy Smiley published "Sense without Denotation" to accommodate logical anomalies within formal logic. (Smiley, 1960)
- Michael Dunn's Ph.D. dissertation provided the semantic basis {T,F,B,N} True, False, Both, None, for Belnap's entailment and the four-valued (numeric) matrix of Timothy Smiley. (Dunn, 1966)
- Kulatissa Jayatilleke published "The Logic of Four Alternatives" (Jayatilleke, 1967) to initiate consideration in formal theory of *tetralemma (catuṣkoṭi)* logic.
- Erwin Chargoff, Maurice Wilkins, Rosalind Franklin, James Watson, and Francis Crick discovered the molecular geometry and tetranary bio-logic structure of data storage and data processing in living cells, based on combinations and permutations of four nucleotides: adenine (A), thymine (T), guanine (G) and cytosine (C), leading to tetranary molecular computing four decades later.

1970s

- Belnap published two papers which shaped the subsequent field of four-valued logic: "A Useful Four-valued Logic" (Belnap, 1977a) and "How a Computer Should Think". (Belnap, 1977b)
- Multiple contributors convened by Belnap and Alan Anderson produce the 540-page Vol. 1 and the 750-page Vol. 2 of '*Entailment: The logic of relevance and necessity'*, which provides the foundations of modern formal four-valued logic. (A. R. Anderson & Belnap, 1975b) (A. R. Anderson & Belnap, 1975a)

1980s

- Graham Priest published "In Contradiction" presenting the rationale of genuine persistent contradictions (*dialetheism*, sentences that can actually be both True AND False), versus merely informational contradictions in the Belnap-Dunn framework (i.e. told True AND told False). He introduced a modern interpretation of ancient *Tetralemma* (*Catușkoți)* logic in which the superposition element 'B' for 'both' signifies active duality: {T,F,B,N} True, False, Both, Neither. This seminal work from 1986 was re-published twenty years later as: (Priest, 2006b)
- Melvin Fitting implemented four-value logic in Prolog. (Fitting, 1988)

1990s

- Leonard Adleman introduced a third method for general-purpose molecular computing based on tetranary logic with the DNA and RNA nucleotides {A,C,G,T} and {A,C,G,U} (Adleman, 1994) (Lipton & Baum, 1996) (Arkin & Endy, 1999).[34] He solved a Hamiltonian path problem (find the shortest route on a network that that visits each vertex exactly once) by associating each vertex with a random 20-mer sequence of DNA, to form a solution of DNA molecules that would encode random paths through the graph. Successive orthogonal filters were then used to sift the optimal path result. This work launched the field of general-purpose *molecular computing*. It which differs in purpose from *gene-based therapeutics* and *materials bio-engineering*. Adelman's work promptly led to the annual *International Conference on DNA Computing and Molecular Programming* which has continued since 1995 (ISNSCE, 2021).

Early 2000s

- Katalin Bimbó and Michael Dunn extended the four-valued approach to general formal logic.

> "It might be controversial that a proposition can be both true and false in a truly ontological sense, but undoubtedly it can be doubly valued epistemically. The simplest example comes from ordinary language usage when one answers a yes-or-no question by "Yes and no." Colliding multiple entries in a database and inconsistent scientific theories are further illustrations of this phenomenon. Classical logic assumes that each proposition has exactly one of the two truth values, 'true' and 'false'. Logics which allow propositions not to have truth values but to have "gaps" were called partial logics; their siblings, logics requiring propositions to be at least true or at least false, possibly having "gluts," were less favored. ... The semantics we provided greatly extends the range of logics which can be given a generalized four-valued semantics." (Bimbó & Dunn, 2001, p. 171-172, 190)

34 Two previous approaches had developed in molecular computing during the 1960s and 1970s. The *bio-molecular* school pursued *n*-ary molecular logic for direct information processing, and whereas the *bio-circuitry* school designed molecular electronics for binary logic with much faster processing speed and memory density. The first used protein geometry for implicit data expression, in complex physical media for dynamic data processing. The second method expressed electronic programs based on binary logic of simple switches built with biomolecular structures configured in series and parallel into explicit symbolic passive circuits with logic gates. (Yates, 1985)(Yates, 1985)

> "The basic concepts in the molecular-computing field go back to the early 1970s, when computing models suitable to biomolecular information processing systems were developed. Since then, researchers have been advancing technical areas that provide the means for fabricating such systems, including biosensors, protein engineering, recombinant DNA technology, polymer chemistry, and artificial membranes." (Conrad, 1986, 55

- General purpose molecular computing based upon tetranary logic with nucleotides of DNA {A,C,G,T} and RNA {A,C,G,U} moved from the lab to early-stage applications: (Benenson et al., 2003) (Damjanović & Rakočević, 2005) (Ignatova et al., 2008) (Mizas et al., 2008) (Phillips & Cardelli, 2009).

2010-20s

- Michael Dunn published the short paper *"Two, Three, Four, Infinity: The Path to the Four-Valued Logic and Beyond"* tracing the roots of multi-valued logic from the classical Indian "Four Corners" *tetralemma* prior to the 6th century B.C.E, through Jan Łukasiewicz' multi-valued framework. (Dunn, 2019)
- Graham Priest published a several English language examinations of *tetralemma* logic (Priest, 2010) (Priest, 2014) (Priest, 2018).
- *Tetralemma* logic was provided a formal logical vocabulary by Takuro Onishi (Onishi, 2015) and a generalized notation by mathematician Giuseppe Greco et. al. (Greco et al., 2019).
- Ravi Madanayake et al. found the four-value *tetralemma* logic data model to be essential for general hypothesis testing:

  > "The original objective of our experimental survey was not to invalidate or undermine Aristotelian Two-Valued Logic, but to evaluate which of the methods ... already used in the published literature .. would be used by our students for the relevant modular transformation. ... According to the results of this experiment, the authors / researchers could conclude that at least in this experiment of this type, the use of Aristotelian two-fold logic (true / false) maybe inadequate to evaluate the validity of a hypotheses. If the Eastern four-fold logic was applied, a result more compatible to reality could be obtained. In that case, a hypotheses could be True, False, Both True and False; or, Neither True nor False." (Madanayake et al., 2015, p. 247)

- Gösta Grahne and Ali Moallemi demonstrated that Belnap's four-valued logic, adapted to relational databases, enables more efficient general-purpose treatment of both incomplete data (NULL) and inconsistent data management. (Grahne & Moallemi, 2018)
- Mohammad Moaiyeri et al. described the use of quaternary logic for nanoelectronics, employing four different voltage levels where {0,1,2,3} means the number of 'thirds of a volt' in functional signals, {$^0/_3$V, $^1/_3$V, $^2/_3$V, $^3/_3$V}, enabling four-valued logic to be implemented on existing binary circuits. (Moaiyeri et al., 2012) Seyyed Ashkan Ebrahimi et al. have demonstrated that this method achieves simpler and more energy-efficient nanotechnology with less information required to save, display or compute, higher computational and transmission speed, more dense storage and memory, reduced system complexity, and easier testing. (Ebrahimi et al., 2016)
- General purpose molecular computing based on tetranary logic with {A,C,G,T} data storage in DNA and {A,C,G,U} data processing via messenger RNA (mRNA) logic gates was formalized and deployed in a variety of fields. This nucleobase method, the bio-molecular method, and the bio-circuitry method are all now integrated into a comprehensive set of molecular computing techniques. (Zhao & Chakrabarty, 2010) (Xie et al., 2010) (Hamano, 2012) (Moon et al., 2012) (Goñi-Moreno & Amos, 2012) (Siuti et al., 2013) (Purcell & Lu, 2014) (Singh, 2014) (Farsad et al., 2016) (Coleman, 2016) (Gaudelli et al., 2017) (Rakočević, 2018) (Dalchau et al., 2018) (Matsuura et al., 2018) (Erbas-Cakmak et al., 2018) (Hong & Šulc, 2019) (Kim et al., 2019) (Spaccasassi et al., 2019) (Wei et al., 2019) (Peng et al., 2020) (Katz, 2020) (Chen et al., 2020) (BioBricks Foundation, 2021) (Chen et al., 2022) (Ma et al., 2022)

Chapter 5: DWDS Technical Rationale and Design Summary

5.1 ‘Data With Direction’ from Concepts to a System Specification

The foregoing chapters identified a general problem of normative communication, sketched a design research methodology to solve it, excavated the foundations of obligation, permission, and encouragement, and then sorted through available technical concepts and methods to determine which of them might be adapted into a capable design.

The present chapter presents a rationale and summary of a solution. A functional map of the new end-to-end system design is provided in Section 5.1 followed by Section 5.2 outlining the purpose of, and progress on, a reference implementation. That is followed by a lengthy Section 5.3 spanning 19 subsections, which incrementally puts shape to the DWDS through functional subtraction, as a sculptor chisels, carves, files and sands away good material in order to leave only the intended object. Sections 5.4 and 5.5 then apply custom data structuring and processing, as the sculptor would add a specialty oil to bring out the lustre in the final work.

The “Data With Direction Specification” (DWDS) describes a type of distributed, general purpose system that individuals and organizations can use to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate rules across any informatics network with precision, simplicity, scale, speed, resilience, and deference to prerogative. DWDS describes a class of data-processing pipeline with the underlying relation: 'IS + RULE ⟹ OUGHT'.

The messages between the rule-maker and taker roles and the RuleReserve network consist of two types of stored data records, and two types transitory data records.

Stored Data

- [rule.dwd]: A normative assertion that uses a tabular input/output logic gate;
- [lookup.dwd]: An ordered list of reference data in two or more ordered columns (an ‘n-tuple’);
  (Unlimited [rule.dwd] and [lookup.dwd] instances can be expressed from a small number of patterns.)

Transitory Data

- [is.dwd]: A request with context and particulars of a circumstance;
- [ought1.dwd]: A response package of [rule.dwd], [lookup.dwd] records with compute tests results.

This specification’s tabular rules-as-data formats are optimized for efficient storage and sifting:

- JSON script to be embedded into or automatically converted into any programming language;
- CBOR virtual logic control table to be called from ToS Bit 6 of an IP packet header (0=No, 1=Yes)

A request will be submitted to the Internet Assigned Numbers Authority (IANA) to register the file extension .dwd (Data With Direction) as a distinct Media Type (formerly called “MIME type”) (Hansen et al., 2005) with available +json and +cbor suffixes (Melnikov & Hansen, 2013).

An individual or entity transmits a structured data *request* message about a circumstance. That message data is used as a sieve to separate out, from a distributed compendium of rules, only those which are ‘in effect’ for the declared jurisdictions and times, and which are ‘applicable’ to the categories declared for that circumstance. The message data is then also used as a different type of sieve on the logic gates of each rule in the shorter list, in order to generate a structured data *response* message that documents what MUST, MAY and SHOULD be relation to that circumstance.

Figure 10 illustrates the general form and flow of DWDS, associating the RuleMaker application with the *imperative* role in normative communication (i.e issuing rules), the RuleReserve network service with the *declarative* role (identifying rules that are ‘in effect’ for a context and ‘applicable’ to a set of categories), and the RuleTaker component with the *empirical* role (sending a set of circumstantial facts and receiving facts about rules deemed to be invoked by the those facts). These distinctions are implicit in other informatics systems for rules from industry (van Dongen & van Maanen, 2013) (SAE, 2018) and government (Mohun & Roberts, 2020) (Schartum, 2016a) (Schartum, 2016b), but they are explicit in DWDS.

- *RuleMaker Agent Role:* The *imperative* prerogative arising from social or institutional spheres empowers a person in the ‘rule-maker agent’ role to compose statutes, contracts, specifications, policies, treaties and other relational structures among two or more parties;
- *RuleReserve Network:* A responsive on-demand request-response data pipeline sifts and transmits *declarative* documentation about rules from a decentralized rule reserve with versioning and redundancy (Cosulschi, 2015)):
- *RuleTaker Agent Role: D*escriptive data about an *empirical* event or change of status is packaged into an [is.dwd] *request* message to the RuleReserve network. When the [ought1.dwd] response message is received back, the logic gates are immediately sifted to produce an [ought2.dwd] result for use by the individual or entity associated with the circumstance, directly or through their machines.

Figure 10: Philosophically-grounded function determines pragmatically-designed form.

NORMATIVE DATA *MUST, MAY and SHOULD*	**Empirical**	**Declarative**	**Imperative**
Informational **Data, Metadata, Schema**	***Fact:*** ***An Event / A Status Change*** (generated / reported / detected) (pending / estimated / potential)	***Communication:*** ***Best Available Information*** (accessible and verifiable) (shaped by relationships)	***Prerogative:*** ***Social or Institutional Agency*** (authority / agreement / preference) (subsidiarity / paramountcy)
Operational *"DWDS"*	[is.dwd] request → ***RuleTaker*** Role ← [ought1.dwd] response	***RuleReserve*** Network	← [rule.dwd] & [lookup.dwd] ***RuleMaker*** Role patterns →
Contextual	***Normative Circumstance*** A set of primary facts invoke some normative propositions, and thus establish a normative circumstance.	***Normative System*** There exists an ensemble of rules which characterize a particular normative order.	***Normative Assertion*** A requirement includes one or more normative propositions.
Practical **MUST, MAY, SHOULD** and their synonyms	***Normative Fact*** or ***Ruled-Based Fact*** A set of primary facts invokes a normative proposition, and therefore establishes the existence of a normative fact.	***Normative Proposition*** or ***Rule Documentation*** There exists a normative proposition relevant to this data which is 'in effect' for this context, and 'applicable' to these facts.	***Norm*** or ***Rule*** Institutional or social norms for *practical* action or status are 'in effect' for a context, and 'applicable' to foreseen facts.
Ethical **MUST, MAY, SHOULD** and their synonyms	***Deontic Fact*** A set of primary facts invokes a normative proposition based on utility, logic, ethics or aesthetics, and thus establishes the existence of a deontic fact.	***Deontic Proposition*** There exists a normative proposition based on utility, logic, ethics or aesthetics 'in effect' for this context, and 'applicable' to these facts.	***Deontic Rule*** Institutional or social views for *ethical* action or status are 'in effect' for a context, and 'applicable' to foreseen facts.

In the pages below, Figures 11 through 14 illustrate with UML Sequence Diagrams the end-to-end data flows that connect the Rule-maker agent and Rule-taker agent Roles, via the RuleReserve Network of a DWDS system. Any node on the Internet may operate all three, or two of, or just one of the three functions.

- *Figure 11 RuleMaker:* A user with social or institutional prerogative for rule-making composes the descriptive expression of a rule, maintains its version history, adds it into a general access reserve file that is disseminated on the Internet, and monitors direct activity for the rule.
- *Figure 12 RuleReserve:* Anyone organizing to set up and operate a Superset RuleReserve node can obtain the latest comprehensive [rulereserve.dwd] catalogue file via its ongoing Internet dissemination; and, anyone running a Subset RuleReserve node can select from that distributed collection. Rule-taker agents may then obtain directly or via their machines, on-demand sifted selections of rules from any node(s) accessible to them across cascading nodes on the network. Downstream end users and nodes can routinely perform automated updates and integrity checks against the audited and certified Superset RR nodes.

- *Figure 13 RuleTaker:* A user agent operating any application to accomplish their purpose may send a request message containing data about a procedural event or change of state, in order to receive back a response message containing sifted, tested data about normative propositions and related control data. Upon being informed of what apparently MUST, MAY and SHOULD be done in that circumstance, the user agent would then scrutinize and choose which part of the response to use.

The term “sift” is chosen to distinguish the operation from “filter”. The two seem synonymous, but are not quite. When something is *filtered*, the user retains the product that passes through the membrane, and discards what was caught in the membrane. When something is *sifted*, the user retains the product caught in the membrane, and discards what passed through sieve. This is an explicit and non-trivial data processing design decision that I and Don Kelly made, based on Robert Kowalski's interpretation of Alfred Horn’s ‘proof-by-contradiction’ expressing all assertions as false, then deriving contradictions in order to deduce which ones are not false (Kowalski, 1979b) (Horn, 1951). This computational strategy enables very fast computer algorithms to test for logical consequence. In contrast, a filter would express all assertions as true, and use contradictions to remove those which are not true.

The default general-purpose peer-to-peer node of a DWDS network incorporates all three RM, RR and RT functions. However any partial combination of these functions can also be useful. For example, devices with minimal or no effective storage capacity can operate as RT-only clients supported by separate dedicated RR servers. This would enable an organization deploying special-purpose equipment to maintain a narrow subset of rules accessible to their installed units. Also RM-only nodes can be employed for the purpose of converting convoluted natural language rules into control tables using simple concise statements in JSON-formatted [rule.dwd] and/or [lookup.dwd] records (JavaScript Object Notation). These can be directly embedded into, or used as pseudo-code for automated transcription into any procedural programming language for operation in other platforms.

The end-to-end DWDS is illustrated in Figure 14 with the example from **Chapter 6** of a dock worker responsible for the management of shipping containers. The agents, their roles and the various actions are summarized with a comprehensive Use Case Diagram overlaid onto the three Sequence Diagrams: Figure 11 (orange), 12 (blue) and 13 (green). The background text is not intended to be readable in Figure 14, rather the purpose of reproducing the Sequence Diagrams here is only to facilitate visual cross-referencing with the full-sized versions. Also it should be noted that the RuleTaker Sequence Diagram is flipped horizontally from the original so that it maps to the right-hand side of the RuleReserve Sequence Diagram.

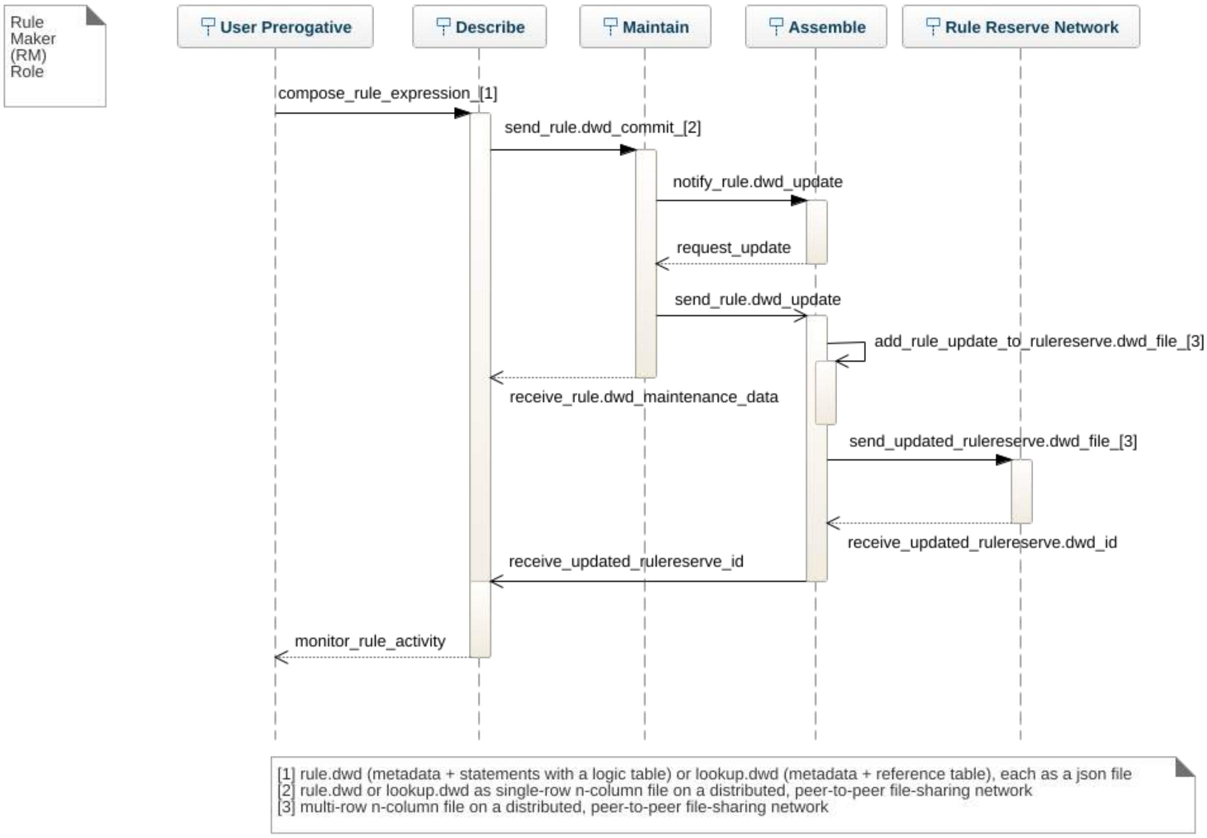


Figure 11: 'RuleMaker Role' Sequence Diagram: Data flows among RuleMaker elements in a DWDS network (an 'Internet of Rules').

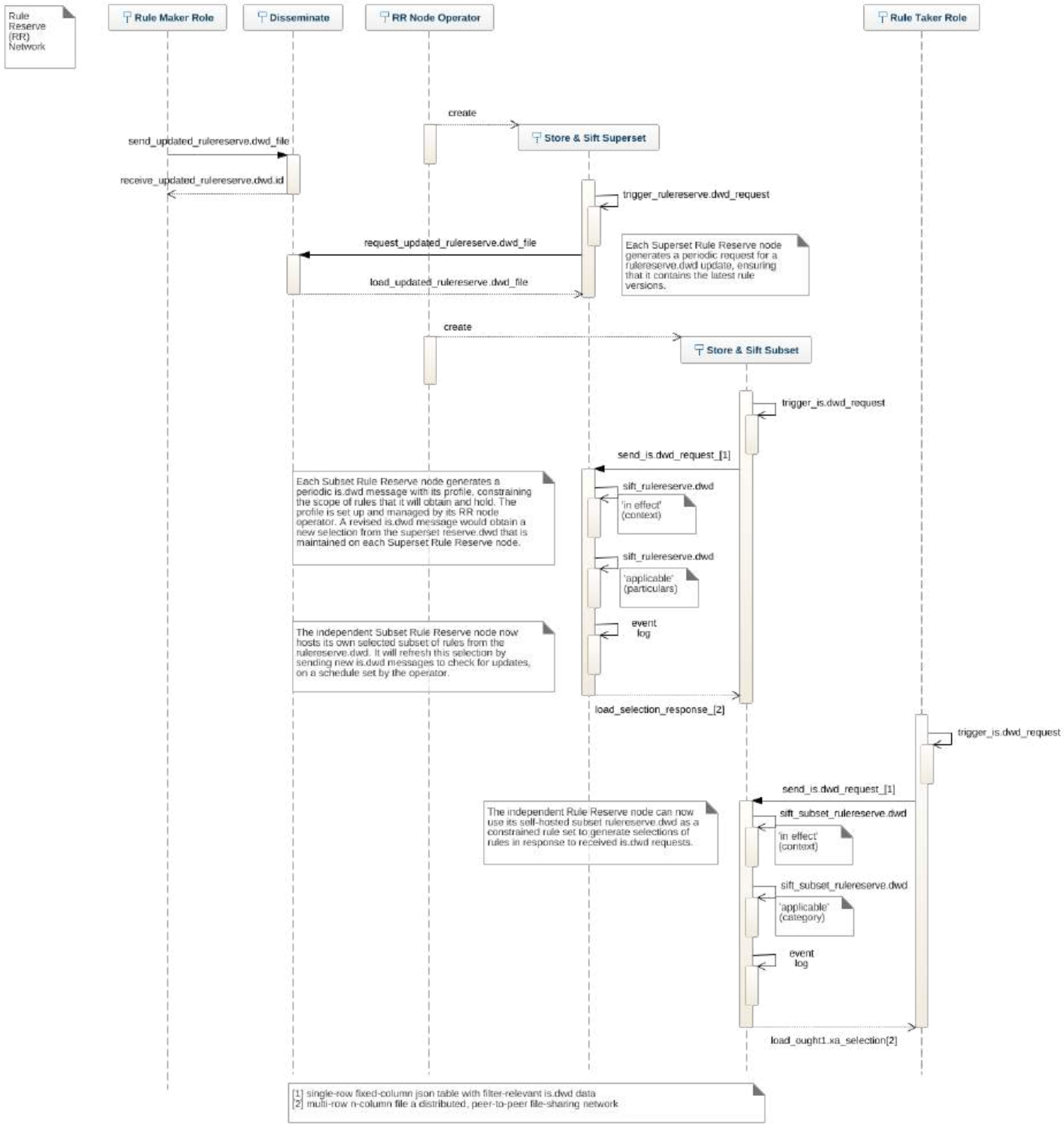


Figure 12: ‘RuleReserve Service’ Sequence Diagram: Data flows among RuleReserve elements in a DWDS network (an ‘Internet of Rules’).

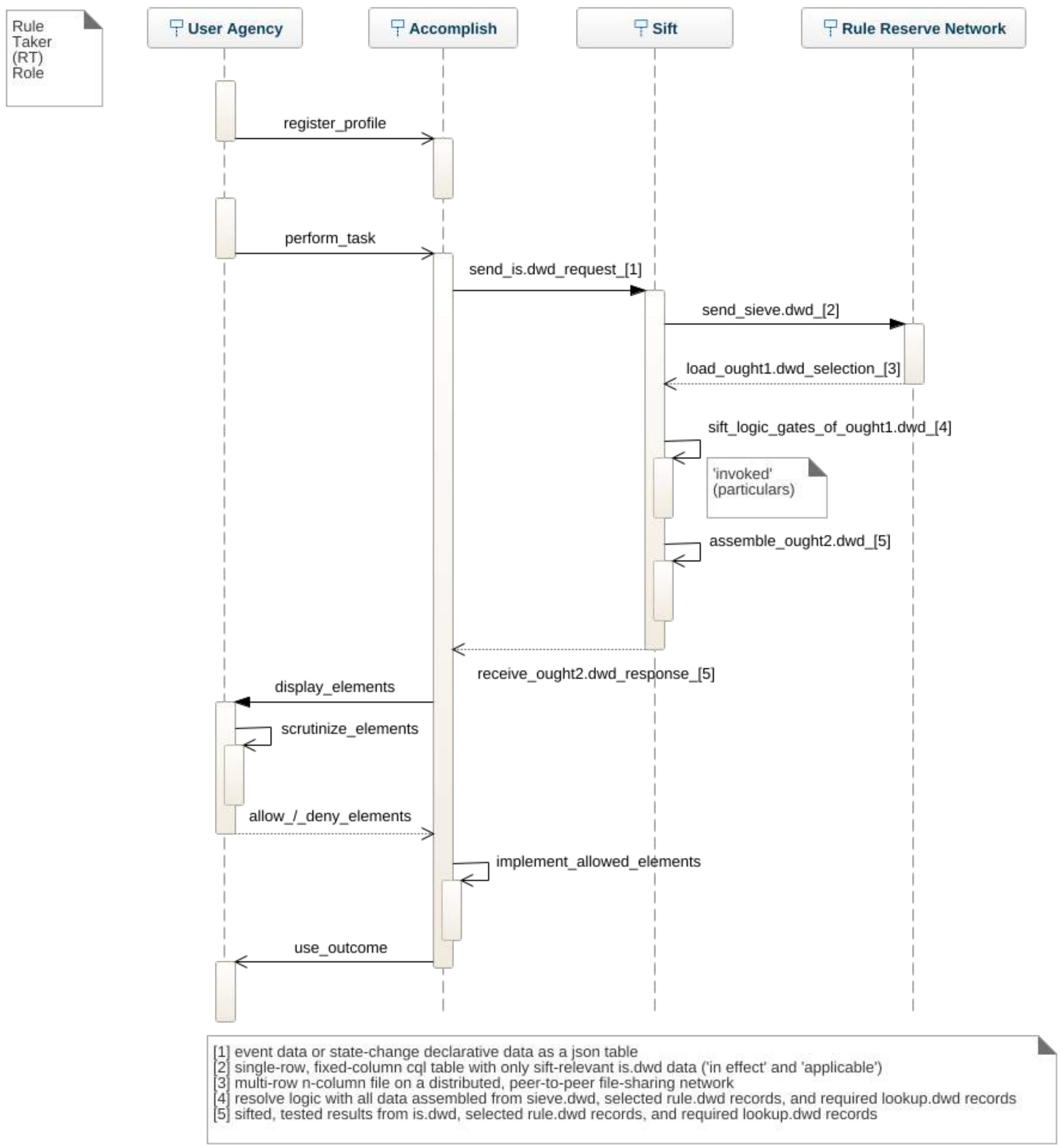


Figure 13: 'RuleTaker Role' Sequence Diagram: Data flows among RuleTaker elements in a DWDS network (an 'Internet of Rules').

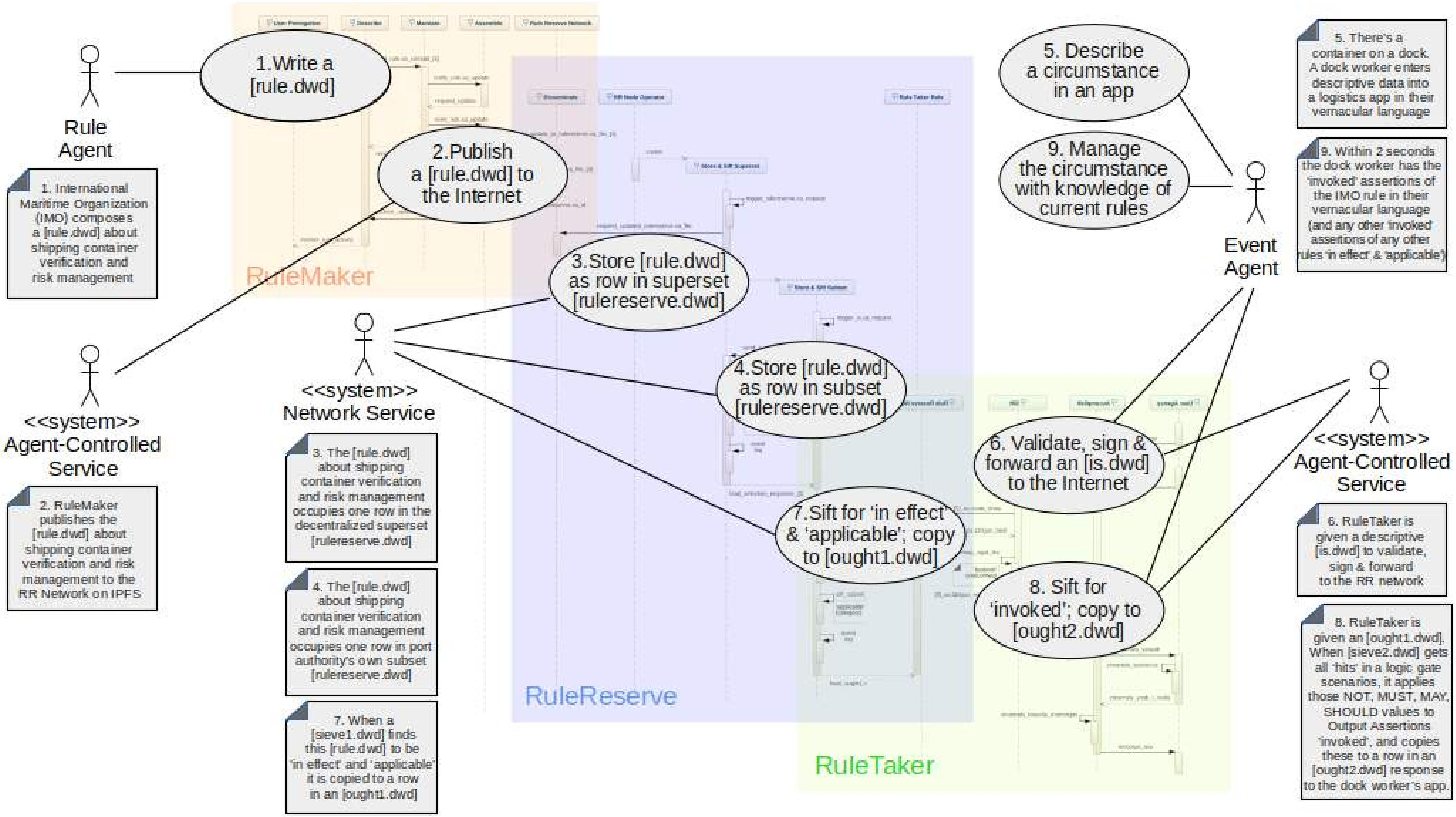


Figure 14: 'An Internet of Rules' Illustrated with a Use Case Diagram Overlaid On RuleMaker, RuleReserve and RuleTaker Sequence Diagrams

5.2 Reference Implementations Under Current Development in Software

DWDS is a general specification that communicates what any implementation of a RuleMaker application, a RuleTaker component, and a RuleReserve network service are supposed to accomplish, without restricting how.[35] Generally speaking any implementation' of the DWD Specification reflects the relation: 'IS + RULE ⟹ OUGHT'. But there are many ways to do that through various particular software packages or services.

A *reference implementation* is a minimal deployment of a design that demonstrates the essential operability of the concept, and provides either a usable pre-built scaffold, or indirect inspiration for a comprehensive *production implementation* by others. Reference implementations usually employ programming languages and platforms that are *optimized for ease of understanding, deployment and maintenance*. Real-world implementations for genuine use would instead be *optimized for precision, scalability, speed and resilience*.[36] Interim implementations help others to understand the original team's intent, and facilitate their adaptation of the design using techniques and technologies appropriate to their own contexts, in their own style, for their own purposes. The design, building and testing of novel elements of a working system, when undertaken iteratively as part of the design research process, helps the designer and participating contributors to think through method coherence and end-to-end composability. To enable a collaborative community, the documentation, programs and source code for the reference implementations are available under widely-used free/libre/open licensing, and they are built entirely with free/libre/open components.

Table 11 summarizes three reference implementations under active development (see Section 6.1 Development of Operational Software Based on the DWDS Design) which illustrate how the DWDS is intended to work. They each have four system elements: RuleData (RD, see Section 5.3.4), RuleMaker (RM), RuleTaker (RT), and RuleReserve (RR). ***Xalgo-rads*** is intended to be capable of adaptation for a genuine production-class Rules-oriented Autonomous Decentralized System (RADS).[37] ***Xalgo-dev*** is a developer-oriented command-line implementation optimized for experimentation, testing and diagnostics. And ***Xalgo-abm*** implements the DWDS in an agent-based modelling (abm) environment for running hypothetical rule scenarios. There may come to be various other implementations of the system, based upon the technical context and specific needs of the implementers.

35 To be precise, a system specification is a set of concepts, definitions, designs, properties, functions, constraints, deployment arrangements, performance objectives, conformance tests and quality assurance methods, that together are detailed enough to be successfully implemented by anyone with general domain knowledge and resources. An informatics system specification must not incorporate restrictions regarding technology platform, programming language, infrastructure, services, suppliers or any other factors that are not intrinsic to the design functions.

36 In high-performance industries, a system specification is only considered to be complete if at least three working implementations meet all the requirements using different technologies on different platforms. (OGC, 2019) (FIDO Alliance, 2020)(FIDO Alliance, 2020)(FIDO Alliance, 2020). The present dissertation is premised upon the only the first implementation, although three are presently foreseen.

37 Kinji Mori led the initial technical conceptualization of Autonomous Decentralized Systems in the 1980s and 1990s. (Mori, 1984)(Mori, 1993)(Mori, 1993) (Mori, 2007)

Table 11: Overview of Reference Implementations of the DWDS

Purposes	Specification	Three Reference Implementations (via Xalgorithms Foundation)		
An Internet of Rules	**DWDS**	**Xalgo-rads**	**Xalgo-dev**	**Xalgo-abm**
RuleData ***Sets syntax for: Metadata; Logic data; Descriptive data.***	***DWDS RuleData*** describes a tabular declarative structure for normative propositions and lookup tables that, when accompanied by a parsing library, can be run directly with, embedded within, or auto-transcribed into, any programming language.	***XalgoRD-rads*** is Xalgorithms Foundation's Internet reference implementation of the DWDS RD specification. Its data model draws upon relevant XML schemas, but all files are expressed with JSON (JavaScript Object Notation) or CBOR (Concise Binary Object Representation).	Specification-conformant DWDS RD runs equivalently in a stand-alone developer environment.	Specification-conformant DWDS RD runs equivalently in an agent-based model.
RuleMaker ***Enables rule authoring, publishing and maintenance. Publishes rules-as-data.***	***DWDS RuleMaker*** describes an easy-to-use application for structured authoring, publishing and maintenance of normative propositions and related reference tables in a local repository that is accessible throughout a DWDS network.	***XalgoRM*** is Xalgorithms Foundation's Internet reference implementation of the DWDS RuleMaker specification (Licensed AGPL 3.0)	***XalgoRM-dev*** is a developer community implementation of the DWDS RuleMaker specification, using a command-line interface (Licensed AGPL 3.0)	***XalgoRM-abm*** is the NetLogo community's agent-based modelling implementation of the DWDS RuleMaker specification. (Licensed AGPL 3.0)
RuleTaker ***Accepts factual data. Sends 'is.dwd'; Receives 'ought1.dwd'; Sifts logic gates to obtain assertions 'invoked'; Presents rules-as-data.***	***DWDS RuleTaker*** describes a library that enables the user of any computing platform to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate normative propositions and related reference tables that have been published on a DWDS network.	***XalgoRT*** is Xalgorithms Foundation's Internet reference implementation of the DWDS RuleTaker library specification. (Licensed Apache 2.0)	***XalgoRT-dev*** is Algorithm Foundation's stand-alone developer implementation of the DWDS RuleTaker library specification, using a command-line interface. (Licensed Apache 2.0)	***XalgoRT-abm*** is the NetLogo community's agent-based modelling implementation of the DWDS RuleTaker library specification. (Licensed Apache 2.0)
RuleReserve ***Receives 'is.dwd'; Sifts metadata to obtain rules 'in effect' and 'applicable'; Sends 'ought1.dwd'.***	***DWDS RuleReserve*** describes a distributed index of metadata routinely updated by every RuleMaker application, and routinely used by every RuleTaker library, to identify, locate and classify all versions of normative propositions and related reference tables on a DWDS network.	***XalgoRR*** is Xalgorithms' reference Internet implementation of the DWDS RuleReserve specification for a distributed index of metadata (Licensed AGPL 3.0)	***XalgoRR-dev*** is Xalgorithms' Foundation's stand-alone developer implementation of the DWDS RuleReserve index (Licensed AGPL 3.0).	***XalgoRR-abm*** is the NetLogo community's agent-based modelling implementation of the DWDS RuleReserve index (Licensed AGPL 3.0).

5.3 Methods for High Performance Decentralized Distributed Computing

5.3.1 Computing Fast and Slow: Externalize Computational Work from Run-Time

Daniel Kahneman's well-known book *Thinking Fast and Slow* is about the distinction in human cognitive psychology between immediate determination and thoughtful deliberation. The two modes have been given various names by different authors: the *System 1 / impulsive / automatic / heuristic* process versus the *System 2 / reflective / reflexive / analytic* process (Kahneman, 2011) (Strack & Deutsch, 2004) (Bargh & Chartrand, 1999) (Evans, 1984). Jonathan Evans first introduced these two modes of cognitive processing as follows:

> "[A] distinction is drawn between two types of thought process which are termed heuristic and analytic. The function of the heuristic process is selection. The outcome of heuristic processing is a judgment of relevance about features of the problem. Information deemed 'irrelevant;' is not processed further. 'Relevant' information is then subjected to analytic processing. ... Suffice to say at present that the function of analytic processes is to generate some form of inference or judgment from the information selected." (Evans, 1984, p. 451)

This difference is also found in the comparison between procedural imperative (‘slow’) versus tabular declarative (‘fast’) data processing with table look-ups. Here in particular we are concerned with fast deterministic data sifting involving the necessary and sufficient information, rather than fast heuristic selection under uncertainty. Simple data sifting methods can be structured so that naïve matching of input symbols against stored symbol combinations in logic tables will rapidly generate the designer’s intended outputs. Numerous deterministic selections in parallel or tree-like configurations may seem to be doing the same thing as slower analytic and inferential methods that require richly expressive programming languages for procedural instructions, parsed and executed through step-by-step sequences (Cunneyworth, 1994) (Vanthienen & Dries, 1993) (Coenen, 1999) (Garcia et al., 2000) (Dean & Ghemawat, 2008b)

This comes down to selecting the correct tool for the job. Using complicated tools for simple jobs is usually feasible, but can be messy: like cutting butter with a chainsaw. Fifteen years ago in the opening of a keynote address entitled *The Computing Machines in the Future,* Richard Feynman characterized the difference between these two data processing styles: “This often goes under the name of artificial intelligence, but I don't like that name. Perhaps the unintelligent machines can do even better than the intelligent ones." (Feynman, 2005, p. 28) We propose to distinguish these as Artificial intelligence (AI) and Artificial naïvety (A∅):

- Artificial intelligence (AI) performs active knowledge acquisition and inductive reasoning with stochastic variational inference and double-loop learning (learning how to learn) to resolve information gaps in order to guide action within a specified degree of risk tolerance.
- Artificial naïvety (A∅) performs passive request-response signal mapping with logic and lookup tables to sift data, and may use basic arithmetic and Boolean operations to transform data. This ‘non-learning’ system retains no user data.

The extreme speed, volume and precision of constrained A∅ data processing can give users the impression of using a very ‘smart’ AI system. But in fact it just involves high-performance ‘dumb’ computing, analogous to: *square peg to square hole—round peg to round hole*.

Several techniques of the DWDS are tailored to reducing the run-time computing burden, with simplicity, speed, precision, resilience and generalizability. There’s not much claim here to ‘innovation’, since each of the techniques described here is long-standing in informatics history, and involves little effort other than to overcome some preconceptions. In particular we anticipate that the primary barrier to adoption would not be due to technological sophistication; instead the challenge is that bright creative informatics systems developers in industry, government and academia don’t typically associate old-fashioned, simple, mostly unused and forgotten techniques with attaining high performance in current-generation applications.[38] It is more natural, understandably, to get excited by new, advanced, big-budget methods (Simons, 2012).

Amid the computing, storage and network infrastructure methods of a half-century ago, some of the brightest creative minds of the 1970s tailored their data structures and data processing techniques to minimize run-time computational resource load, while maximizing generated information utility (information with an actionable purpose is *knowledge*). Far-seeing management thinkers at that time, such as Adrian McDonough, Deryke Belshaw and Theodore Levitt, anticipated the practical problems that arise with too much information. (McDonough, 1963, p. 72) (Belshaw, 1981) (Levitt, 1991, p. 6) Their advice was to design information systems that are:

- *'Cost effective':* If equivalent output can be made based on reduced data and resources, then less ought to be acquired and committed; and
- *'Cost efficient':* Improvements in output performance ought to be justifiable in view of the data input collection, management and computational burden.

Into the 1990s and 2000s easier programming methods facilitated much more programming functionality, lines of code and complexity, and this could be accommodated because more rapidly advancing computer hardware and network performance could handle the heavier run-time computational loads and data storage requirements. But the down side of that progress was to accommodate less efficient, less effective data structures and data processing methods.

Now in the 2020s, the proliferation of data, devices and demand brings back the need for informational efficiency and effectiveness: we've got a new iteration of that 1970s-generation challenge. How can we use the least data processing resources to support the best decisions?

Just as there are certain domain-specific designs and methods which are homologous to the implementation of any high-speed train, any emergency response service, or any top-tier athlete, we suggest that a passion for informational efficiency and effectiveness are necessary to attain the virtues, norms and overall character of the intended design of DWDS. There's certainly no prohibition against forking our design, other than that in principle, one should not claim or imply that an inconsistent project is an implementation of the 'DWDS' specification.[39]

38 Shaker Zahra and Gerard George define absorptive capacity as the routines and processes by which people and their organizations "acquire, assimilate, transform, and exploit knowledge". These factors determine how individuals and entities dynamically evolve in their own methods of operation, and also in relation to others. When people have the motivation, resources and time to make use of assimilated knowledge, then as individuals and in their organizational roles, they tend to be more receptive to acquiring and assimilating that new knowledge. (Zahra & George, 2002)

39 See the classic essay of eXtreme Programming by Ron Jeffries: "We Tried Baseball and It Didn't Work". (Jeffries, 2006)

The following sections explain various techniques incorporated into this specification which, in each case, externalize work. Each of these techniques is conceptually independent, so they are discussed separately. However there's an emergent synergy attainable when they are combined into a single method, in the complex manner Christopher Alexander explains in *"A Timeless Way of Building"* (Alexander, 1979).

5.3.2 Transforming Complex Natural Language to a Simple Controlled Natural Language

A rule author using an implementation of the RuleMaker specification need not write any programming code. The RuleMaker interface helps the user express simple normative propositions in uniformly-controlled natural language, even though they may originate in unstructured natural language. The Input Conditions and Output Assertions are stated in a consistent type of declarative sentence, and the logical relations among them are structured into an adjacent table.

The first pre-requisite to converting unstructured free-form natural language into uniformly-controlled natural language is to compose clear sentences. This can seem obvious to the point of condescending, but such refinement is often difficult to achieve in written expression when there is need for precision, or when the subject is complicated or complex. The Internet Engineering Task force specifies: " 'Simplification of language' here refers to ways of controlling expressions in a language to make reading or comprehension easier for particular target audiences" (Phillips & Davis, 2009) In the 1950s the UK Government had Ernest Gowers provide guidance in *Plain Words* for how to achieve straightforward communication, as this is indispensable to getting practical work done:

> "But what is this job that must be got on with? ... the writer's job is to make his reader apprehend his meaning readily and precisely. ... Even when he knows what he means, and says it in a way that is clear to him, is it always equally clear to his reader? If not, he has not been getting on with the job." (Gowers, 1954, p. 78)

DWDS RuleData (RD) is a data specification for the purpose of transmitting normative propositions among rule-maker agents and rule-taker agents across a network. It is based on a limited phrase-structure grammar with just six syntactic elements and one declarative mode, which we shall referred to here as RuleFiniteStateGrammar (RFSG). Despite this very narrow set of syntactic constraints, there are no boundaries on semantic scope, nor on the order in which the syntactic elements can be used. This is inverse to the more common Semantic Web technique of supporting complex expression using rigid semantic schemas (e.g. RuleML) and tolerant syntactic structures (e.g. SGML). The two approaches are not mutually exclusive; they are complementary in the sense that controlled vocabularies can be used within the controlled syntactic elements of RuleMaker application.

The DWDS framework leaves the management of semantics in the hands of people who have the prerogative, motivation, domain knowledge and socio-cultural familiarity to tailor the expression of each sentence of each rule, and who are motivated to make a genuine effort to provide a faithful reproduction of the full normative intent of the original rule, with minimal distortion.

Once a rule is comprehensibly and accurately expressed in the controlled natural language of DWDS RuleData, involving sentences structured with RuleFiniteStateGrammar in a logic gate, and combined with the required metadata and optional descriptors of this specification, it is then suitable to be published to the Internet in a way that it can be easily discovered, fetched, scrutinized, prioritized and, with agreement of direct stakeholders, automated with devices over a network.

Anyone provided a few examples of sentences in RuleFiniteStateGrammar, in any natural language that they understand, should be able to author rules correctly in that language or in any other natural language they are fluent in. Although this can be time-consuming due to the need to focus on the exact meanings of the normative propositions, this work can be tackled by having many people with moderate domain knowledge thoughtfully choosing and configuring words and phrases to compose the six elements of the DWDS template, and arranging these elements in a sequence that makes sense in their working language. In any language it is intuitive to identify each syntactic segment of each sentence.

However the theoretical and technical aspect of controlled natural language specifications is an advanced theme in the field of linguistics. There are many different types of controlled natural language frameworks and it will be useful to situate the present DWDS among them. Tobias Kuhn offers "A Survey and Classification of Controlled Natural Languages" in which he characterizes a hundred of them with a descriptive scheme he called PENS. This is an acronym based on four distinguishing characteristics: *precision*; *expressiveness*; *naturalness* and *simplicity*. In his approach these characteristics are rated 1 to 5 meaning *none, low, medium high maximal* (Kuhn, 2014, p. 7, 22). Kuhn also suggests a categorization scheme with nine descriptors:

- Function:
  - C—comprehensibility
  - T—translation
  - F—formalization
- Media:
  - W—written
  - S—spoken
- Terminology:
  - D—domain-specific
  - A—academia
  - I—industry
  - G—government

Based on these characteristics Kuhn provides a very useful annotated bibliography. However I suggest that it would also be useful to distinguish controlled language frameworks by their functional purpose: *'What does one do with them?'*. Such a typology is summarized in Table 12, in which two main groups are (a) controlled natural language frameworks for the purpose of writing rules; and (b) controlled natural language frameworks for the purpose of analyzing rules.

This primary distinction between controlled natural language frameworks ‘for writing’ or ‘for analyzing’ occurred to me during the October 2022 “Rules-as-Data Workshop” hosted by Alessia Damonte and Giulia Bazzan. (Damonte & Bazzan, 2022) This event mainly involved contributors to the “Institutional Grammar 2.0” community, (Frantz & Siddiki, 2022) for whom the “rules as data” phrase refers to a body of published rules used as an empirical data source to analyze the institutions issuing them.

> “The Institutional Grammar supports syntactic and semantic classification of the rules that embed within regulatory text. A recently revised version of the Institutional Grammar – Institutional Grammar 2.0 – further supports robust and reliable coding of regulatory text by supporting: (i) comprehensive representations of rules that embed in regulatory text along a generalizable syntax; (ii) accommodating coding of the heterogenous forms of rules often encountered in regulatory language; and (iii) supporting computational analysis and modeling of regulatory rules in real world and simulated settings.” (Frantz & Siddiki, 2022, p. 1)

This analytical purpose differs from the meaning of Jan Vanthienen who explains that a “decision table contains rules — rules as data ... expressed as rows or columns" with a functional dependency. "Given a decision table with conditions and actions, a set of actions is functionally dependent on a set of conditions if every combination of condition values corresponds to (one and) only one configuration of action values." (Vanthienen, 2010)

In a similar manner as Vanthienen describes, I and a community of colleagues through Xalgorithms Foundation have been using the “rules-as-data” phrase to refer to writing or transcribing rules into data structures that are easy to store, discover and transmit across a distributed network, and easy to parse within any functional application. For example as Craig Atkinson and Nicolás Schubert write (citing my dissertation (Potvin, 2023)) : “Chile aims to become the first jurisdiction to express and publish trade rules as ‘standardisable’ data packages (i.e., ‘Rules as Data’) that can be readily fetched and utilised by any person using any software application on any platform. Participants in the collaboration foresee these efforts as an incremental step towards the emergence of a what can be described as an ‘Internet of Rules’ (IoR).” (Atkinson & Schubert, 2021) See also: (Mohun & Roberts, 2020, p. 48-49)

Any general typology is, of course, a gross simplification, inevitably unsatisfactory in its details. But I consider that this one will be successful if it leads to shared reflection and conversation in order to elaborate and refine my initial characterizations in an attempt to distinguish these types and examples.

Table 12: A proposed categorization of 'Controlled Natural Language' frameworks suitable for writing and analyzing rules. The four columns to the right are loosely adapted from (Kuhn, 2014). Some modifications are due to comments on an earlier draft from Christopher Frantz (Frantz, personal communication, January 8, 2023)

CONTROLLED NATURAL LANGUAGE FRAMEWORK	EXAMPLE	PURPOSE	RESOURCE	FOCUS	PRECISION	EXPRESS-IVENESS	NATURALNESS	SIMPLICITY
Types of CNL Frameworks for Writing Rules								
SEMANTIC	Semantics of Business Vocab-ulary and Rules	standardize the vocabulary for a domain	meta-model meta-rules	institutional language	high	medium	high	high
SYNTACTIC	Augmented Backus-Naur Form	define the structure for a language	grammar	precise syntax	high	medium	medium	medium
SEMANTIC & SYNTACTIC	Logical Form	define vocabu-lary & structure for a language	logic	logical expression	high	medium	medium	medium
Types of CNL Frameworks for Analyzing Rules								
STRUCTURAL	Institutional Grammar 2.0	parse & interpret meaning in statements	syntax and logic	institutional statements	medium	high	high	high
PRAGMATIC	Grice's Maxims	parse & interpret meaning in discourse	principles	conversational language	medium	high	high	high

Three types controlled natural language framework are portrayed under the first primary group and two are shown under the second, though additional types and sub-types can be added. This arrangement distinguishes the frameworks according to: their *purpose*; the *resources* they employ; and their *focus* of attention. Kuhn's four PENS characteristics are included, but here only a "high / medium / low" rating is used instead of his five numeric indicators. The frameworks referred to in the table are summarized below, along with with source references to the examples:

Controlled Natural Language Frameworks Intended for Writing Rules

- *Semantic Controlled Natural Language* e.g. Semantics of Business Vocabulary and Rules (SBVR) (OMG, 2005) standardizes the meaning of words and phrases to reduce communication ambiguity.
- *Syntactic Controlled Natural Language e.g. Augmented Backus-Naur Form (ABNF)* (Overell & Crocker, 2008) is focused on structural elements and rules of language.
- *Semantic-Syntactic Controlled Natural Language* e.g. Logical Form (Frege, 1879) combines both semantic and syntactic rules to construct meaning with words and sentence structure.[41]

Controlled Natural Language Frameworks Intended for Analyzing Rules

- *Structural Controlled Natural Language* e.g. Institutional Grammar 2.0 (Frantz & Siddiki, 2022) is used to analyze the syntax and phrases of statements (one or more sentences) to determine meaning.
- *Pragmatic Controlled Natural Language* e.g. Grice's Maxims (Grice, 1975) is used to analyze how language is used in discourse to convey certain meanings in particular contexts.

41 Among the hundred examples summarized by Kuhn, my approach somewhat reflects the intent and approach found in "ClearTalk", (Kuhn, 2014, 29-30)(Kuhn, 2014, 29-30)(Kuhn, 2014, 29-30) which is detailed in Doug Skuce's informally published paper "*A Controlled Language for Knowledge Formulation on the Semantic Web*" (Skuce, 2003). It is based on the same author's doctoral thesis from 25 years earlier: *"Towards Communicating Qualitative Knowledge Between Scientists and Machines"* (Skuce, 1977). In my typology ClearTalk fits into the Semantic-Syntactic category. Its purpose is to help users find sentences expressing certain semantic types of facts about certain concepts, and Skuce does this by including semantics within the language, whereas DWDS design accomplishes this with metadata.

In the context of this named typology, the present DWDS design puts forward a new type of syntactic controlled natural language framework *for writing rules*. DWDS relies on a purely context free “finite state grammar” inspired by Augmented Backus-Naur Form (ABNF) which is listed as the representative example provided in Table 12.[42] A context-free grammar provides a set of syntactic elements and some rules about how they can be combined. It can require that certain syntactic elements be limited to words and phrases that have particular grammatical roles (e.g. nouns; adverbs; articles). There are no constraints on the semantic meanings of words and phrases. By not requiring a *‘domain-specific language’* (DSL), RuleData accommodates unlimited semantic scope.

The syntactic structure of sentences expressible in the logic table is narrowly constrained. Consequently rule authors will sometimes find that a sentence they want to express in a particular natural language is an awkward fit within our set of six syntactic elements. However they can re-arrange the order of the elements displayed, because the syntactic relationships of the elements is retained even when the observed sequence changes. Rule authors can include connecting words such as prepositions or articles into any of the fields, along with punctuation. Trade-offs will sometimes be required, but practical meaning should generally take precedence over elegance of expression.

The two primary groups of controlled natural language frameworks proposed here, ‘for writing’ versus ‘for analyzing’, should be seen as complementary. This use of a controlled natural language framework for *upstream rule writing* can result in *downstream rules analysis* that involves less time and complexity, and fewer assumptions, for improved accuracy, precision, and completeness. Specifically, when the texts have been composed with limited and explicit syntactic elements, based on consistent sentence templates, and clear meta-rules for the use of part of speech (i.e. nouns, verbs, prepositions, etc.), logical operators (i.e. arithmetic; Boolean; logical from) and specific notations, then natural language parsing and interpretation can be greatly simplified.

5.3.3 Externalize Linguistic Complexity from Rule Structure, to Simplify Function

The DWDS involves explicit decomposition of *syntactic structure* and *logic structure* from *semantic expression*. Run-time computational work can then be focused upon simple, efficient matching of symbols. Semantic complexity is externalized from run-time by having rule-makers place natural language words into a pre-determined set of syntactic fields with fixed grammatical functions. Together these elements enable the construction of just one type of declarative sentence, but any number and variation of such sentences can then be positioned as row labels of a table that supplies the logic structure and the normative mode of a rule. This method is explained in detail below.

42 Backus-Naur Form is mentioned only indirectly in Kuhn’s review, as it is a context-free grammar rather than language.

5.3.3.1 *Making the Logic Relations Explicit*

A DWDS logic table employs one sentence per row of a tetranary vertical I/O table, relating each input condition and each output assertion to {00,01,10,11} symbols. In this manner, the logic structure of a normative proposition is easy to see, even when it is relatively complicated. This juxtaposition of each proposition with its potential states was described in the 1960s by organizational theorist Myron Tribus:

> "All the messages sent by the observer, insofar as they pertain to scientific observations, may be represented in a two-part coded system. The first part may be called the substantive and the second the interpretive part of the message. ... The substantive part is an unambiguous statement, in English (if that is the observer's native language), in the form of a proposition. The other part of the message is an interpretation of the truth value or credibility of the substantive part of the message. It is the observer's way of interpreting the subject matter without pretending to be omniscient about the subject. … The substantive part of the message specifies something about which to be uncertain. ... A simple symbolic method of communication for our observer-receiver pair is a notation such as p (A|X) , where A represents a proposition, X represents the conceptual framework within which the meaning of A is to be understood, and p (A|X) represents the numerical interpretation of the credibility of A." (Tribus et al., 1966, p. 245)

5.3.3.2 *Making the Syntactic Elements Explicit*

In natural language communication, humans imply and infer syntactic function by context, without needing each element to be explicitly named in the message. In contrast computer programs require explicit identification of the *syntactic* structure of each *semantic* expression, something which John Backus formalized with *Backus Normal Form* (BNF) (Backus et al., 1960).[43]

In computational tables it is routine to structure information by using the individual names of either the syntactic elements or semantic classes as the labels of rows or columns. DWDS logic tables can employ complete sentences as the row labels in a vertically stacked topology, but each sentence consists of a fixed set of syntactic elements and relations, in whatever order is required by the language used.

Brian Roark and Richard Sproat comment that "Grammars built for computational syntactic processing must typically trade-off the richness of syntactic description provided by the grammar with the computational cost of using it." (Roark & Sproat, 2007, p. `17) In the DWDS we reduce the computational burden by constraining expressiveness to a set of just six *syntactic* elements. Only some of the six elements are required for the simple computational operations, but the full set is required in the form of a sentence to support human comprehension.

43 'Backus Normal Form' (BNF) evolved over the years with improved compactness, simplicity and representational power, resulting in 'Augmented BNF' (ABNF) (Overell & Crocker, 2008). The general technique of explicitly pairing a functional signal to each semantic expression became routine across programming languages, including for semantic metadata tagging of data in Standard Generalized Markup Language (SGML) (ISO, 1986). This practice enables interoperability across otherwise heterogeneous platforms, and thus, systems modularity.

When considered in relation to the conventional methodology of mathematical linguistics underlying most computer programming logic, our design is pursued in the reverse direction:

- A linguist observes many sentences of a language and infers a finite set of syntactic rules that anyone may use to assemble words of that language to convey infinite semantic meanings.
- We design a single finite syntactic pattern for declarative sentences and use this in a structured set for anyone to assemble words of any language to convey infinite semantic meanings.

Every row label of the DWDS logic table contains a grammatically-equivalent statement. This rigid sentence template provides a consistent language-agnostic set of grammatically consistent syntactic *elements* and *relations*. An arbitrary limit of 240 characters per sentence reduces malicious potential, and encourages a concise style. Although each element of a row is displayed for human reading as a sentence, the elements themselves are stored as distinct data fields with known attributes. This pre-parsed semantic data can be processed efficiently.

In *Syntactic Structures*, Noam Chomsky emphasizes the general utility of a kernel of basic sentences within "a simple or context-free phrase-structure grammar":[44]

> "A finite state grammar is the simplest type of grammar which, with a finite amount of apparatus, can generate an infinite number of sentences." ... "The central idea behind transformational analysis is that it will be profitable to select among grammatical sentences a certain kernel of basic sentences for which a simple phrase structure can be described, and in which all grammatical relations and selectional relations can be discovered." ... "We can greatly simplify the description of English and gain new and important insight into its formal structure if we limit the direct description in terms of phrase structure to a kernel of basic sentences (simple, declarative, active, with no complex verb or noun phrases), deriving all other sentences from these (more properly, from the strings that underlie them) by transformation, possibly repeated. (Chomsky, 1957, p. 24, 82, 106-107)

Chomsky suggested the utility of a small kernel of basic sentences that are "independent of meaning" (Chomsky, 2000, 17, 125). The DWDS logic table takes this a step further by reducing the small kernel of sentences to a fixed-grammar sentence template with just six syntactic elements that can be arranged to create particular declarative sentences. Despite the rigid limitation, this template has infinite semantic range because it can be populated with words from any lexicon, language and character set.

The context-free "typed feature structure grammar" (Wintner & Sarkar, 2002) of a sentence in the phrase-structure of DWDS RuleFiniteStateGrammar contains the following elements (represented in ABNF):

44 Chomsky comments: "It is interesting to observe, in this connection, that the theory of context-free phrase-structure grammar is very close to adequate for 'artificial languages' invented for various purposes, for example, for mathematics or logic or as computer languages." (Chomsky, 2000, p.138 fn 36)As far back as the 1950s he observed that "one possible method for describing a grammar is in terms of a program for a universal Turing machine. The purpose of this paper is to examine the properties of such a grammar and language for a finite state machine. (Chomsky, 1958, p. 92)

DWDS Sentence =

Determiner = the | this | these | a
SubjectNoun (word or phrase)
PastParticipleVerb (word or phrase referring to the subject noun)
AuxiliaryVerb = be | do | have (optionally with NOT)
ObjectNounOrVerb (optional with a preposition)
ObjectDescriptor (optionally an adjective, adverb or arithmetic)

The structure of a sample sentence using this template can be described with "Minimal Recursion Semantics" (MRS), a framework for computational grammar decomposition introduced by Ann Copestake (Copestake et al., 2005) (Copestake, 2002). In Figure 15 we use the online “Linguistic Knowledge Builder” (LKB) (Carroll et al., 2020) application to visually illustrate one of the declarative sentences from the earlier example of a grocery store delivery policy: *“This box as measured is the standard type.”*

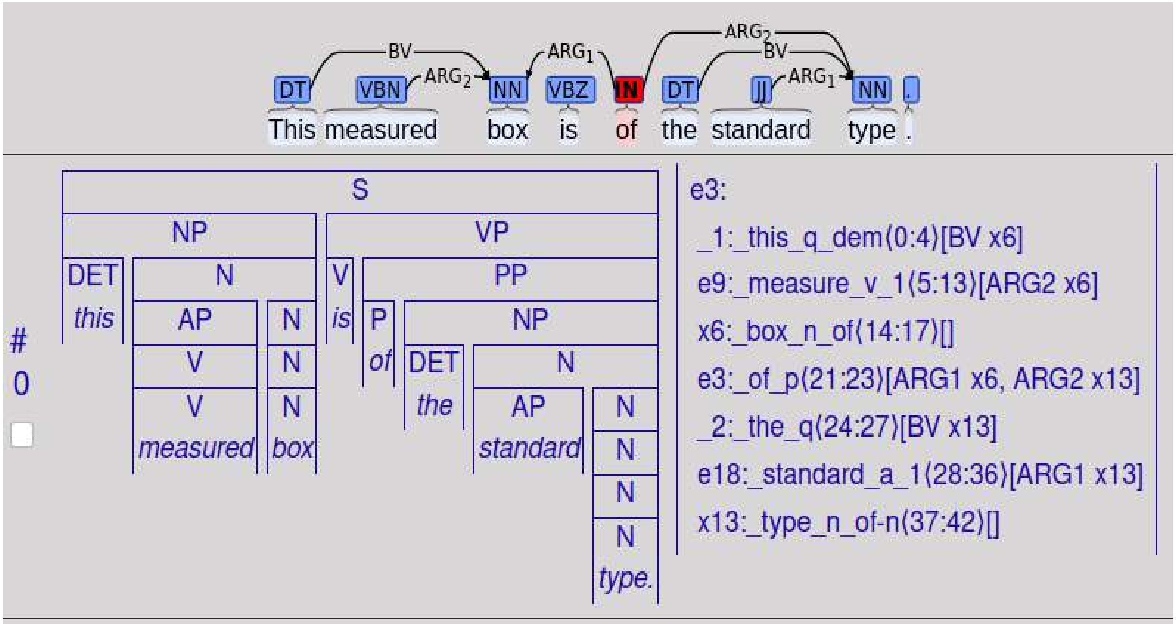


Figure 15: Syntax Structure of a Sentence in the DWDS RuleFiniteStateGrammar, within the RuleData Model

Prepared with *Linguistic Knowledge Builder (LKB)*, a free/libre/open source grammar analytics application for typed feature structures (Carroll et al., 2020). http://erg.delph-in.net/logon

<u>KEY</u>
DT: determiner;
VBN: verb past participle;
VBZ: verbAuxiliary;
NN: noun;
IN: preposition;
MD: modal (normative verb)
JJ: adjective|arithmetic;
BV: bound variable;
ARG: argument

5.3.3.3 *Making the Syntactic Structure Explicit*

The MRS analysis shows that the sentence structure employed for DWDS logic tables combines two assertions (grammatical ‘arguments’), each of which expresses a subject-predicate-object relation:

> *ARG_1: This box type is standard*; and,
> *ARG_2: This box type is measured.*

Together these may be expressed as: *“This box as measured is the standard type.”*[45] Table 13 has a colour-coded list of the six syntactic element categories of the DWDS sentence template, and the two subject-predicate-object arguments they express. The information characterizing the main assertion *ARG_1* (*This box type is standard)* has a black background; and the information of the supporting assertion *ARG_2* (*This box type is measured)* is shown with a white background. Each element contains a word or phrase to perform the named function.

Table 13: Syntactic Elements of a Sample Sentence in the DWDS RuleFiniteStateGrammar, within the DWDS RuleData Model

Syntactic Element	Required vs. Optional	May be Combined With	Sample Sentence
Determiner	*Optional*		This
SubjectNoun	*Required*		box
PastParticipleVerb	*Required*	*SubjectNoun*	as measured
AuxiliaryVerb	*Optional*		is
Object Descriptor	*Required*	*ObjectDescriptor*	the standard
ObjectNounOrVerb	*Optional*	*ObjectNounOrVerb*	type.

To check variations in syntactic structure with these elements, an arbitrary sample of acquaintances of the lead researcher was informally asked to interpret the sample English statement into their own first language. The translators were not provided any other grammatical instructions or wider context; they were just asked if they would please share back how to express the sample sentence in the other language that they were fluent in.[46] Some very helpful people reached out to their friends and provided several additional language versions.

45 The present draft of this dissertation describes a more recent version of the DWDS logic table design than is implemented in the software component used to generate the image in Figure 19.

46 Some translators pointed out a flaw in the sample sentence. One explained as follows: “According to my own thinking: When we measure something, this has to do with its size, not its type. When we speak for type, we examine the object's specifications. So, I find the phrase: *‘This box, as measured, is of standard size’* or *‘This box, as examined, is of standard type’* more reasonable.” Certainly the word "type" should be "size". The observation is correct, and the inelegant sample wording was initially an oversight. But we retained it to learn how translators would handle the categorital term "type".

The test described here was not intended to be a formal part of this design research, rather it began as just an informal review with various colleagues to validate a technical detail for logical coherence. It will come as no surprise to a linguist that this evolved into an interesting step on a linguistic journey, resulting in the valuable insights documented in this section.[47]

The general template for sentences in the logic table of DWDS RuleData is designed with the intention of accommodating expression in any natural language using any character set, with elements displayed with the orientation, order and flow direction of the language in use (e.g. horizontal left-to-right; horizontal right-to-left; vertical top-to-bottom) (W3C, 2019). Table 14 provides a small sample of translations of the one sample sentence arranged and coloured for comparison. (Languages which are conventionally vertical such as Japanese, Mandarin and Korean are displayed here horizontally.)

The coloured sequences in Tables 14 and 15 portray the *syntactic structures* provided by each person. Our impromptu unstructured experiment suggests that adequate universality can be obtained with a common set of syntactic elements, albeit requiring some flexibility. A template for sentences in the DWDS must accommodate user-determined arrangements of the syntactic elements, while retaining the semantic relationships amongst them. Evidently there is no need to instruct rule authors about linguistic theory; it appears to be sufficient to provide them an example or two of well-formed sentences with the intended syntactic structure for a language they are comfortable with.

This structure illustrates a general syntactic 'vehicle' suited to the design intent of the DWDS, but we need to accommodate the varying syntactic structures of diverse languages to achieve the same class of statement in any language. Some specific observations from this test are:

- The 'Determiner' (the, this, these, a) is not always used;
- Two elements might be combined into one compound word (Finnish: **vakiotyyppi**)
- Some elements may be implicit rather than explicit (e.g. Myanmar, Russian, Hungarian)
- A linguist informed us that some languages do not have declarative subject-predicate-object triples, as they do not have a 'subject' element. Our design can substitute another element for this.

47 Multi-language translation was not initially intended as a formal part of this design research. It was just a ‘due diligence’ check for integrity. Machine translation was not used because it cannot assure the fluency and precision we were after.

Table 14: Syntax Structure of the Same Sample Sentence in Various Natural Languages

Shona: **Bhokisi iri sekuyerwa kwara kaitwa ndere mwero wepakati.**

Luganda: **Eno bokisi nga bwe'tugipimye ya ngeli sitandadi.**

Indonesian: **Kotak ini seperti diukur adalah tipe standar.**

Portugese: **Essa caixa, que foi medida, é do tipo padrão.**

Polish: **To zmierzone pudelko jest standardowego typu.**

Dutch: **Deze doos zoals gemeten is van type standaard.**

Lingala: **Linzanza oyo emekelo ezali oyo e longobami.**

English: **This box as measured is of the standard type.**

Greek: Αυτό το κουτί, όπως μετρήθηκε, είναι κανονικού τύπου.

Croatian: **Ova izmerena kutija je standardnog tipa.**

Finnish: **Tämä laatikko mitattuna on vakiotyyppi.**

Ukrainian: Цей блок, як виміряно, є стандартного типу.

Russian: Эта коробка, по размерам, стандартного типа.

Serbian: Ова измерена кутија се стандардног типа.

French: **Cette boîte mesurée est de type standard.**

Spanish: **Esta caja medida es de tipo estándar.**

Swahili: **Sanduku hili, hupimwa, ni aina ya kiwango.**

Japanese: √ 이측이정된상자는표준형입니다す。

Hungarian: **Ez a doboz normal tipus méretű.**

Myanmar: ဒီ ဗူးက အတိုင်းအရ ပုံမှန်ပဲ။.

Mandarin: 这杀子按测量是标准格式。

Korean: 이 측정 된 상자는 표준형입니다.

Thai: กลล่องนนี้ตามวัดเปปืนแบบมาตรฐาน

Arabic: هذه العلبة كما تم قياسها فهي من النوع الأساسي

Hebrew: הכופסה הזאת כנמדד היא סוג סטנד

Urdu: یہ باکس قسم معیاری پیمانے کے مطابق ہے

Table 15: Syntax Sequences from a Sample Sentence in Various Natural Languages

English:		**Luganda:**	
Japanese:		**Dutch:**	
Mandarin:		**French:**	
Finnish:		**Spanish:**	
Portugese:		**Korean:**	
Greek:		**Polish:**	
Shona:		**Thai:**	
Croatian:		**Indonesian:**	
Serbian:		**Lingala:**	
Russian:		**Swahili:**	
Myanmar:		**Hebrew:**	
Hungarian:		**Arabic:**	
		Urdu:	

An application using this template will need to be tolerant of various connector words and punctuation that may accompany natural language expression, while still being being optimized for decentralized storage, rapid discovery and efficient computation. To illustrate, the sentence "**This measured box is a standard type**." is operationally equivalent to "**This box as measured is the standard type**." Both of these would carry the same operational effect as if someone with limited capability in English were to say: "**The box, measured, is type standard**." This is also flexible to alternative expressions, for example in Portuguese: **"Essa caixa, que foi medida, é do tipo padrão."** can be equivalently expressed as **"Essa caixa, conforme as medidas, é do tipo padrão."**

Within each language there is typically more than one syntactic structural variant available for the same basic sentence. Syntactic variability and semantic dissonance within and among languages complicate natural language processing. While of course some semantic meaning can be lost based on particular words[48] and due to different syntactic sequencing, the general incentive of a conventional user of the RuleMaker application is to authentically communicate the accurate meaning in each language. People can make mistakes, but the difficulty of semantic alignment is greatly reduced in the DWDS RuleFiniteStateGrammar by working with only one sentence template that always contains the same six syntactic elements to build all sentences in all logic tables. Semantic variability can be handled more simply as communities of rule-maker agents and rule-taker agents have a shared interest to develop and choose standard schemas. Any application relying upon this structure can provide a way for users to access synonyms and multiple languages through [lookup.dwd] reference tables.

48 Three of our informal translators sought contextual clarification whether the word 'box' was meant as a physical container, or as a data entry field in an administrative form. Also certain words in this sample sentence are out of context in some languages. In the Namuy Wam language of Cauca (southwestern Colombia), there is no term for "a box" (in Spanish "una caja"). So if the intent is to refer to a moderately large package for cargo, a functional synonym in Namuy Wam is "un costal" (a large heavy-duty sack).

5.3.3.4 *Making Rules Easily Readable and Efficiently Computable*

The DWDS provides a structure for ‘rules-as-data’ that is directly usable by non-specialized humans and by general-purpose computers. A rule author using a RuleMaker implementation supplies simultaneously:

1. A rule in their chosen natural language, in simple human-readable form;
2. A syntactically pre-parsed data package in machine-processable form.

Following is our sample sentence written in JSON, and associated with the logic relations using the symbols {1,10,0} to be discussed later. Each of the syntactic elements of the sentence comprises a field with a value.

When the sentences of the DWDS logic gates are composed in applications conformant to the RuleMaker specification, they are pre-parsed to the consistent syntactic structure of the RuleFiniteStateGrammar. This can be displayed in JSON as shown at right, but the logic gate data is actually stored and processed like a horizontal tape within the single row occupied by the [rule.dwd] record of the decentralized [rulereserve.dwd] database. This horizontal topology is also employed for rules-as-data in [ought1.dwd] and [ought2.dwd] response messages transmitted across an Internet of Rules, which are described later.

```
"sentence": {
 "Determiner": "This",
 "PastParticipleVerb": "as measured",
 "SubjectNoun": "box,
 "AuxiliaryVerb": "is",
 "Object Descriptor: "the standard",
 "ObjectNounOrVerb": "type"
 }
 {
 "cases": [
 {
 "case": "A",
 "value": "00"
 },
 {
 "case": "B",
 "value": "01"
 },
 {
 "case": "C",
 "value": "01"
 }
 ],
 }
```

This information can be audited directly for integrity, but it is also trivial to display in a browser or application interface in a way that a human can read a properly-flowing sentence. The named elements of syntactic structure (i.e the fields such as "Determiner" and "SubjectNoun") can be displayed to human readers in any order that makes sense to them. And it is trivial for a program or database to employ particular elements for any computational purpose.

In the course of active DWDS computation, the sentences of the logic gates in [rule.dwd] records expressed as shown above are sifted and processed with structured [is.dwd] messages. Selected elements create subject-predicate-object triples (SPOTs) in one-layered or two-layered relation. The underlined elements are required, and the others are used if present:

""<u>SubjectNoun</u>”:“PastParticipleVerb"" : ""ObjectNounOrVerb”:“<u>ObjectDescriptor</u>""

For example the computational parts of our sample sentence may contain the following data:

""box":"measured"" : ""type":"standard""

This minimalist expressiveness will function with any natural language, upon the working premise that when an element is missing it can be considered implicit or combined. To test this, we performed a 'naïve' rudimentary test. The present researcher does not understand the Myanmar language and did not use any translation-assembled elements from the sample Myanmar sentence supplied by a professional linguist for the earlier test. The following was built by relying solely on the known role of each syntactic elements, without considering the semantics at all:

""ဗူးက" : "အတိုင်းအရ""” : "ပုံမှန်ပဲ။"

The linguist confirmed that the essential semantic meaning was retained in this limited data. Our objective is that the same should be workable with the sentence data from Arabic, read right-to-left. JSON syntax works equivalently in both directions.

" الخانة كم " : "" اا تم قياسها" : "فهييأساسية""

These examples show how circumstantial data can be pre-structured in [is.dwd] messages to be used in constructing transitory [sieve1.dwd] packages to sift rules-as-data stored within RuleReserve nodes. It is similarly straightforward to auto construct *field:value* combinations for the response [ought1.dwd] message sent back to the RuleTaker node.

Any sentence built with the six syntactic elements of the DWDS RuleFiniteStateGrammar can usually be reduced to four or three or two of those syntactic elements without loss of communication utility, despite the obvious reduction in stylistic elegance. The RuleMaker application has a minimum subject-predicate-object triple involving the <u>SubjectNoun</u> and <u>Object Descriptor</u>. But any rule author composing a sentence in the RuleMaker application can mark either or both of the PastParticipleVerb and ObjectNounOrVerb fields as essential to their intended meaning, for the purpose of coherent RuleTaker sifting operations. Following are some examples:

Example (a):

```
 "sentence": {
 "Determiner":"This",
 "SubjectNoun":"container status,
 "PastParticipleVerb":"is listed as",
 "AuxiliaryVerb":"being",
 "ObjectNounOrVerb":"considered",
 "Object Descriptor:"loaded"
 }
```

Two-layered computable relation:

```
""container_status":"is_listed_as"" : ""considered":"loaded""
```

One-layered computable relation :

```
"container_status":"loaded"
```

Example (b):

```
 "sentence": {
 "Determiner":"This",
 "SubjectNoun":"box,
 "PastParticipleVerb":"as measured",
 "AuxiliaryVerb":"is of",
 "ObjectNounOrVerb":"type",
 "Object Descriptor:"standard"
 }
```

Two-layered computable relation:

```
""box":"measured"" : ""type":"standard""
```

One-layered computable relation:

```
"box":"standard"
```

Example (c):

```
 "sentence": {
 "Determiner":"The",
 "SubjectNoun":"number of residential properties",
 "PastParticipleVerb":"currently registered",
 "ObjectNounOrVerb":"to the purchaser",
 "AuxiliaryVerb":"is",
 "Object Descriptor:"2"
 }
```

Two-layered computable relation:

```
""number_of_residential_properties":"currently_registered"" : ""to_the_purchaser":"2""
```

One-layered computable relation:

```
"number_of_residential_properties":"2"
```

Example (d):

```
"sentence": {
"Determiner":"The”,
"SubjectNoun":"country of residence",
"PastParticipleVerb":"deemed habitual",
"ObjectNounOrVerb":"of the person",
"AuxiliaryVerb":"is",
"Object Descriptor:"Chile”
}
```

Two-layered computable relation:

```
""country_of_residence":"deemed_habitual"" : ""of the person":"Chile""
```

One-layered computable relation:

```
"country_of_residence":"Chile"
```

These examples demonstrate the expressive tolerance of the DWDS RuleFiniteStateGrammar and of the overall end-to-end system that employs it. This has advantages, but it can also cause problems. Bringing together words and phrases from disparate users in all sorts of contexts, using different languages and dialects, seems to leave far too much to chance. The distributed decentralized composition of [rule.dwd] records, and transmission of circumstantial data in [is.dwd] messages from independently-operated RuleTaker components, would seem to create the conditions for a very level of data matching which the sifting method depends upon. Exact equivalency of terms can hardly be taken for granted among diverse autonomous users across an open distributed network.

On the other hand, anyone taking the trouble to employ the RuleMaker and RuleTaker applications has a straightforward incentive to optimize their own provision data in a manner that would generate reliable and accurate results. There are two primary reasons to expect a sufficient incentive for interoperability to emerge:

- *Utility of Alignment to Semantic Standards:* The DWDS can operate without standardized lexicons, but certainly many communities of rule-maker agents and rule-taker agents will have shared interests in the increased efficiency and precision of using and extending the engagement of standard schemas. Specialized communities of rule-makers and rule- takers tend to require or prefer standardized terms. To the extent rule authors and the designers of application data structures become aware of and acquire an interest in having rule logic auto-generated in the background in an intelligible way, it seems reasonable to expect that they will be attracted by the efficiencies easily obtained by migrating towards a shared lexicon, such as Semantic Web data schemas that have been negotiated over the past quarter century across many specialized communities of practice. (Berners-Lee et al., 2001) What does not come to be uniformly standardized due to prior implementation of incompatible data models, can be handled through automated data mapping.
- *Use of Lookups for Data Mapping:* Tolerance of expressive diversity across the user community can be actively facilitated through a variety of techniques, one of which involves drawing upon [lookup.dwd] tables of synonymous words and phrases. Data mapping lookups can accommodate multi-language expression, as well as two or more overlapping schemas employing different categories and terms (e.g. UN/CEFACT and UBL). (Parry et al., 2010)

> In this way, data can ‘correspond’ without having to ‘match’ characters, for example the English `"box":"standard"` corresponds with the Spanish `"caja": "estándarizado"`, and with proximate strings to accommodate different expressive styles, as `"caja":"estándar"`. Each rule-maker agent and rule-taker agent can retain the autonomy to use the lexicon and language they prefer, even where semantic standards are well established. They can list any alternative parsing resource as [lookup.dwd] tables in the [rule.dwd] metadata, and/or in their own RuleTaker configuration settings.

It remains to provide a simple method for keeping organized all the syntactic structures organized across any of the languages that rule-maker agents and rule-taker agents may employ for rule expression. The DWDS enables the individual syntactic elements of each sentence to be kept organized for display to human readers. This is done with a simple numeric [lookup.dwd] record that relates languages of expression to the syntactic structures of the sentence template used by all rule authors for all rules in all languages. It would be maintained just as any other table of reference data on an Internet of Rules.

A mandatory field presented in the authoring form of the RuleMaker application requires that the user indicates the language of the rule sentences they are expressing. This attaches language metadata to each [rule.dwd] file. The method by which the DWDS accommodates all written languages is through rudimentary data structuring.

The language identifiers lookup table is generalized to be entirely numeric (in Western Arabic numerals) by performing two simple data substitutions: languages are represented with ISO standard 3-letter codes expressed in ASCII decimal form; and, syntactic elements are represented with arbitrary W3C standard numeric colour codes.

To specify the language of each [rule.dwd] record we rely on ISO 639-2:1998 *“Codes for the representation of names of languages”* shown in Table 16. (ISO, 2016) To generate a numeric representation for our operational lookup table, we arbitrarily make use of the ASCII decimal codes that correspond to each three-letter string (e.g. 102,114,97 instead of fra for Français). With the above codes and sequence information we can assemble an extensible numeric [lookup.dwd] file for the syntactic structure in any natural language for the sentence structures used in DWDS logic tables. This tabular data is easy to process in any computational application on any platform written in any programming language, and also easy to render in any preferred way through a graphical user interface.[49] A similar rudimentary data substitution is employed for the syntactic elements. In this case we arbitrarily make use of standard numeric RGB codes (Çelik et al., 2018) corresponding to the colours that were used to illustrate the syntactic structure in our present document. This convenient data substitution provides distinguishable strings of digits, and the colours designated by these codes can be used to set colours to syntactic elements in Web and application interfaces on any platform.

49 This table is not intended for direct viewing by general users, though it is intended to facilitate error-free auditing by humans. A system auditor could easily overlook Mandarin Chinese being represented with the code *man*, but the standard actually specifies *chi* and *zho*. Such errors of expectation are reduced by using numeric strings.

Table 16: ISO Codes for Natural Languages

English name of Language	ISO639-2	ISO639-2 as ASCII Code
French	fra	102,114,97
Dutch	dut	100,117,116
Indonesian	ind	105,110,100
Lugandan	lug	108,117,103
Spanish	spa	115,112,97
English	eng	101,110,103
Polish	pol	112,111,108
Korean	kor	107,111,114
Japanese	jpn	106,112,110
Thai	tha	116,104,97
Hebrew	heb	104,101,98

Table 17: RGB Colour Codes Used as Syntactic Element Identifiers in the DWDS

Color	RGB Code[50]	Syntactic Elements for Sentences in a DWDS Logic Table
	0,0,0	determiner (the-this-these-a)
	255,0,0	subject noun (word-phrase)
	0,0,255	past participle referring to the subject noun (word-phrase)
	0,255,255	verb (be-do-have / optional: must-may-should / optional: not)
	0,255,0	object (noun-verb / optional: preposition)
	255,255,0	object (adjective-adverb-arithmetic)

To the extent people using a particular language choose a common lexicon in that language for use in these six fields, this method enables tabular programming directly in any natural language. A rule author writing in a particular language can arrange the sentences of a logic table for the rule in the appropriate grammatical sequence in the user interface. The [lookup.dwd] data in Table 18 represents the syntax sequences that were illustrated in Table 15, using numeric codes for the colours illustrated in Table 17. This enables every [rule.dwd] file to be grammatically pre-parsed for transmission and storage. This helps to minimize the computational work required for automated rules processing by any platform

50 The *W3C standard for colors* expresses three-part Red-Green-Blue (RGB) numeric codes using 0 to 255, where 0,0,0 diplays as black and 255,255,255 displays as white. This is because in programs made for electronic displays, these values are converted into voltage intensities sent to red, green and blue light sources. The reason for the value 255 is that this is the highest number that can be expressed with one byte (8 bits binary integers, either a 0 or a 1). There are 2^8 = 256 combinations expressible in one byte of binary data. The number 0 in 8-bit binary is 00000000, while 255 in 8-bit binary is 11111111. The expressible range for each coloured light source voltage intensity is 0-255, which enables 256^3 = 16,777,216 discrete combinations, or perceived colours. It might be useful to assign the RGB color codes where syntactic elements are comfined. For example the green and yellow of the Finnish compound word for standard type would be (128, 255, 0). However the utility of doing so has not yet been thought through.

at the edge. A node of the DWDS RuleReserve network can deliver this rule as a grammatically pre-parsed JSON or CBOR file, which can be read by human directly in their own language, and is relatively easy to work with in any applications on any computing platform.

Table 18: Position from Center of the Six Syntactic Elements in Sentences of a DWDS Logic Table

ISO639-2 as ASCII Code	**-3**	**-2**	**-1**	**1**	**2**	**3**
102,114,97	0,0,0	255,0,0	0,0,255	0,255,255	0,255,0	255,255,0
100,117,116	0,0,0	255,0,0	0,0,255	0,255,255	0,255,0	255,255,0
105,110,100	0,0,0	255,0,0	0,0,255	0,255,255	0,255,0	255,255,0
108,117,103	0,0,0	255,0,0	0,0,255	0,255,255	0,255,0	255,255,0
115,112,97	0,0,0	255,0,0	0,0,255	0,255,255	0,255,0	255,255,0
101,110,103	0,0,0	255,0,0	0,0,255	0,255,255	255,255,0	0,255,0
112,111,108	0,0,0	0,0,255	255,0,0	0,255,255	255,255,0	0,255,0
107,111,114	0,0,0	0,0,255	255,0,0	0,255,255	255,255,0	0,255,0
106,112,110	0,0,0	255,0,0	0,0,255	0,255,255	255,255,0	0,255,0
116,104,97	255,0,0	0,0,0	0,0,255	0,255,255	0,255,0	255,255,0
104,101,98	255,255,0	0,255,0	0,255,255	0,0,255	0,0,0	255,0,0

5.3.4 Externalize Engagement of Semantic Web Standards to Rule Makers and Rule Takers

Whereas a Semantic Web schema *constrains semantic expression while accommodating unlimited syntactic structures,* the DWDS schema *constrains syntactic structure while accommodating unlimited semantic expression.* The sentences of the DWDS logic gate have a “finite state grammar” as described in Noam Chomsky’s earliest work: *Syntactic Structures*. (Chomsky, 1957) This achieves the dual purpose of (a) greatly simplifying any machine parsing of the sentences, and (b) leaving all semantic control with the rule authors. Rule authors can embed controlled data from standard semantic schemas into the sentences of DWDS logic gates. This approach ensures that the resulting Internet of Rules is compatible (does not conflict) with any semantic schemas that specialized user communities choose to engage. It provides an incentive for rule makers and rule takers to tend towards common standard terms and schemas, and provides a way to use look-up tables to reconcile incompatibilities among different schemas. This approach to integration with Semantic Web standards arose though communications with leaders of various semantic standards.[51]

51 In particular, thanks go to Ken Holman, Ron Ross and the late Harold Boley.

5.3.5 Externalize Computability by Requiring Rule Expression to be NOT Turing-Complete

DWDS RuleData is designed for tuple-oriented programming (Underwood, 2011), with the mandatory requirement that it remain less than required for Turing-complete expression. There are several reasons for this:

1. Turing-completeness is not necessary to solve the class of problem addressed by DWDS, therefore risks that come with Turing-completeness are intrinsically pre-empted, and bounded.
2. DWDS RuleData supports strictly-declarative computing. A rule may take time to compute, but inspection can validate that it will halt. The "halting problem" is germane to Turing-complete procedural computing.
3. The open distributed nature of a DWDS system requires intrinsic design tactics to reduce the risk that DWDS message injection attacks may introduce procedural logic into user environments.

Each of the DWDS records and message types is packaged with a common metadata set to facilitate discovery, transmission, composability and audit. But each record is a stand-alone entity, because RuleData is unable to express the invocation of rules within rules. Any record can express the fact of dependency upon other records, but this is only “incorporation by reference” without any way to automatically call them, nor even to recursively call itself. There is no specification-conformant capability within RulesSchema, or in the design or the RuleReserve nodes to express the injection of a procedural method or instruction, or a similar symbol for a rule, and have it compute within a RuleReserve or RuleTaker implementation differently than an original stored rule.

When another rule, lookup table or symbol is referred to, the assembly of the two or more elements is performed entirely by the RuleReserve nodes, and these nodes are capable of parsing only a very short list of characters, patterns and functions limited to simple deterministic state transformation operations (Gurevich, 2014) without any provision for looping constructs (recursors, iteration) (Moschovakis, 2001). Each RuleReserve node has just enough operational capability to perform naïve data sifting in order to select rules that are ‘in effect’ and ‘applicable’ to a circumstance as documented in an [is.dwd] message.

The DWDS requires that transmitted data must be network-transitory, that’s to say it must not be stored or copied for retention by operators of the network. Hence, there’s no massive data storage facility for problematic files to hide. The rule-taker agents, rule-maker agents and repository operators have the responsibility to retain their own auditable activity logs.

It is understood that whether intentionally or in error, Turing-completeness can creep into any non-Turing-complete specification as various developers, immersed in thinking about and solving particular problems, may overlook whole system composability. For this reason, ongoing systems analysis is essential to maintaining this strict design constraint.

When a use-case does require additional programming capabilities, it is acceptable within the DWDS RuleData specification for a rule author to include a pointer to one or more external references where the required procedural code can be obtained. But the RuleReserve and RuleTaker interpreters MUST NOT have the ability to acquire or execute those external functions. On the other hand, there is no interference with an end users' liberty to configure their own systems to obtain, validate and run those external program elements within their own control and risk-tolerance parameters. The functional scope of the DWDS is auxiliary to computing prerogatives at the edge.

5.3.6 Externalize Control Data and Logical Relations Data by Separating Data from Procedure

In *Structure and Interpretation of Computer Programs* (commonly known as "*The Wizard Book*" since its publication in 1984) Harold Abelson with Jerry and Julie Sussman refer to:

> ..."the general distinction between describing properties of things and describing how to do things, or, as it is sometimes referred to, the distinction between declarative knowledge and imperative knowledge. In mathematics we are usually concerned with declarative (what is) descriptions,whereas in computer science we are usually concerned with imperative (how to) descriptions. ... [M]uch of the technical difficulty of this subject has to do with negotiating the transition between imperative statements (from which programs are constructed) and declarative statements (which can be used to deduce things)." (Abelson et al., 1984, Sec. 1.1.7)

As earlier described through CODASYL, and implemented in Prolog, they suggest to automatically produce 'how to' from 'what is':

> "The idea is to make interpreters sophisticated enough so that, given 'what is' knowledge specified by the programmer, they can generate 'how to' knowledge automatically. This cannot be done in general, but there are important areas where progress has been made."

In the DWDS we also separate 'what is' data[52] from 'how to' procedures:

- Logical relations data and control data: GIVEN "a" is; WHEN "b" is; THEN "c" is;
- Imperative procedural logic operations. IF *"d"* is*;* THEN do "e*";* ELSE do "f*"*.

52 Defined as: ***Data*** ≡ semantic signal;
Information ≡ data + contextual comprehension; and,
Knowledge ≡ information + actionable purpose

Each of these is an elemental structure of logical triangulation.[53]

- Declarative Data Triangulation: *Two Inputs—One Output:*

  GIVEN-WHEN-THEN logic statement relates an *empirical* context [GIVEN *'a'* is] AND *empirical* circumstance [WHEN *'b'* is], with a consequent *declarative* proposition [THEN *'c'* is]. (North, 2006) (Fowler, 2013).

- Imperative Procedure Triangulation: *One Input—Two Outputs*:

  An IF-THEN-ELSE logic statement relates a *contingent* future state [IF *'a'* is], with either a consequent *imperative* action [THEN do *'b'*] , OR [IF *'a'* is not], a default *imperative* action [ELSE do *'c'*]. (Mladenic et al., 2003, p. 8-9)

The 'structural integrity' of a computational system is obtained through the use of these two elementary units. They can be difficult to discern, because even in relatively simple programs they may be combined into sequences and/or be running in parallel and/or be nested into each other.[54] Should a programmer inadvertently omit just one vertex of one such triangular unit, that program will 'fall over'. However whole units can be externalized, such as by leaving end users to supply the logical relations data. The programmer could write a set of "non-deterministic" *imperative procedures* that are open-ended until the user supplies a set of *declarative* logical relations data and control data (such can be obtains from an Internet of Rules). Abelson and the Sussmans distinguish between such modular non-deterministic programming, versus the common deterministic style in which programmers assert end-to-end control:

> [M]ost programming languages are strongly biased toward unidirectional computations (computations with well-defined inputs and outputs). There are, however, radically different programming languages that relax this bias. ... In a nondeterministic language, expressions can have more than one value, and, as a result, the computation is dealing with relations rather than with single-valued functions. Part of the power comes from the fact that a single "what is" fact can be used to solve a number of different problems that would have different "how to" components. ... The nondeterministic program evaluator ... moves away from the view that programming is about constructing algorithms for computing unidirectional functions. ... Logic programming extends this idea by combining a relational vision of programming with a powerful kind of symbolic pattern matching. ... [T]he user must choose from the set of mathematically equivalent networks a suitable network to specify a particular computation. ... This approach, when it works, can be a very powerful way to write programs. (Re-arranged from: Abelson et al., 1984, Sec. 1.1.7)

53 Triangulation is most commonly referred to in geographical deduction such as land surveying, celestial navigation, Global Positioning Satellite (GPS) services and telecommunications cell tower operations. A target's exact location can be pinpointed on a map with reference to any other two observable points. *Omnitriangulation* is the generalized logic structure described by Buckminster Fuller: "By structure, we mean a self-stabilizing pattern. The triangle is the only self-stabilizing polygon. By structure we mean omnitriangulated. The triangle is the only structure. ... [A]ny recognized patterns are inherently recognizable only by virtue of their triangularly structured pattern integrities. ... Only triangularly structured patterns are regenerative patterns. Triangular structuring is pattern integrity itself. This is what we mean by structure." (Fuller, 1975, Sec. 610.00 Triangulation)(Fuller, 1975, Sec. 610.00 Triangulation)(Fuller, 1975, Sec. 610.00 Triangulation)(Fuller, 1975, Sec. 610.00 Triangulation)

54 Multi-value CASE expression is an IF-THEN-ELSE logic structure with ordered lists (tuples) in which inputs "a1, a2, .. , aN" and outputs "b1, b2, .. , bN" supply sets of values for the same variable classes "a" and "b". Although CASE programming uses the term WHEN, this is really is a contingent IF because it needs to be terminated with an ELSE.

```
IF "a"
  THEN "b"
ELSE "c"
```

```
IF "a1, a2, .. , aN"
  THEN "b1, b2, .. , bN"
ELSE "c"
```

The DWDS design can achieve an order of magnitude gain in simplicity because domain specialists compose the sentences and graphically set the logic relations, and *programmers no longer have to interpret or write the rule logic* because:

- The RuleMaker program automatically generates and publishes well-formed JSON script for the syntactically-structured sentences and for the tabular numeric logic, and stores each complete rule in a single row in RuleReserve network.
- In RuleTaker, all of the logic of each rule known to be ‘in effect’ and ‘applicable’ is received in a row of the [ought1.dwd] message from the RuleReserve. Immediately upon arrival the [is.dwd] data is used to sift the logic gates of the ought.dwd, leaving only the assertions to be invoked.

Software programmers implementing DWDS do not have to interpret, write or resolve any rule logic. They are free instead to focus on creating elegant functions and interfaces to use and display the results. Then all that’s left is maintenance and monitoring.

An important corollary of this observation is that domain specialists with a deep understanding of the nuanced rules of their respective fields, who want to partially or fully automate them, will no longer need to hand over the interpretation of their rules to software programmers. They are able to express advanced and nuanced rule logic with the user-friendly RuleMaker interface.

All logical relations data, control data, and decisions within the operational scope of the DWDS are externalized to end-user human agents who perform rule-maker and rule-taker roles, directly or through their machines. Programmers have the role of writing the underlying procedural components that transmit and evaluate the data which users choose to provide.

This approach to separation of procedure from data is a respectful concession to end user prerogative in the spirit of “human-centred design” (Graeber & Billings, 1989) (Mitchell, 1996), and it also sets the stage for computational systems with a structural integrity that is easy to write, maintain and audit.

5.3.7 Externalize the Data Processing Burden with Purposeful Structuring of Data Into Tables

5.3.7.1 Data Topology Overview

Structuring information into simple two-dimensional tables is literally as old as the wheel.[55] As a method, the ordinary table of data is commonly overlooked as a labour-saving invention in its own right. The structuring of information into tabular form seems so utterly obvious that 'it goes without saying'. But tables of data for representation and structured operations can liberate a person to concentrate upon whatever primary task they are undertaking, without digressions:

- An engineer designing a machine quickly checks a table of dimensions for hexagonal nuts;
- An analyst comparing a study group to a control group checks a table for the relevant '*t statistic*'.

Every trifling lookup to a reference table enables the user to externalize the concentrated work of the many specialists who conceptualized a data model, debated the sources and methods, categorized, processed and assembled results into each cell, checked for errors, and made the data set available.

It is easy to overlook available efficiencies that tabular computing can attain, compared with step-wise computing from input data with procedural code. Informatic systems optimized for in-memory input/output computing with tabular logic and data structures can be orders of magnitude faster and more efficient than procedural imperative methods running on the processors.

Tables can also make the logic structure of a program more comprehensible. Art Lew observed that a program "can be designed as a decision table, be executed as is, and be self-documenting!" (Lew, 1983, p. 183) An input/output table is straightforward, and many types of flaws can be noticed with a cursory glance. By comparison, a procedural program of nested IF-THEN-ELSE sequences is sometimes easier to write in step-wise fashion, and to read line-by-line. But it leaves error detection relatively more difficult. Consequently sometimes auditors of large procedural programs painstakingly transcribe procedural code into tabular declarative form, just to grasp what it's doing. (Janicki et al., 1997) (Janicki, 1995) (Parnas et al., 1994)

The *Data-Driven Transformation* of the past 70 years (Zdankus & Delli Colli, 2021) (Dubois et al., 2000) includes tabular declarative computing; for example, for the beginning it has been routine to use tables for reference data such as identifiers, categories, indices and addressing is routine in computing. But in our view, the potential of a decentralized global network of computationally consistent tabular logic gates has yet to be realized. The DWDS is our contribution to this pursuit.

55 Martin Campbell-Kelly et.al. explain: "Tables have been with us for some 4500 years. For at least the last two millennia they have been the main calculation aid, and in dynamic form remain important today. Their importance as a central component and generator of scientific advance over that period can be underestimated by sheer familiarity." (Campbell-Kelly et al., 2004, p. 1)(Campbell-Kelly et al., 2004, p. 1)(Campbell-Kelly et al., 2004, p. 1) Our current-generation electronic databases and spreadsheets implement the same essential tabular technique as was used to structure trade and commerce annotations on the ancient Mesopotamian clay tablets (Schmandt-Besserat, 2009), and decimal multiplication tables on bamboo strips that pre-dated China's unification (Qiu, 2014), The system of Roman numerals depended on lookup tables for undertaking practical calculation. (Maher & Makowski, 2001, p. 382-383)

The DWDS logic gate employs three elemental topologies for tabular data, each sharing the same schema: GIVEN a set of contextual facts WHEN a set (minumum two) of particular facts are also documented, THEN some normative are deemed to be 'in effect' and 'applicable'. All three table topologies are shown in Table 19, each requiring two or more input conditions.

Cartesian Product

GIVEN	WHEN
WHEN	THEN

Vertical Stack

GIVEN
WHEN
WHEN
THEN

Horizontal Tape

GIVEN	WHEN	WHEN	THEN

Table 19: Three General Data Topologies with a Consistent Declarative Data Scheme

Upon reviewing the data topologies of Table 19, Don Kelly commented that DWDS seem like a 'core kernel' specification that reminds him concepts underlying the kernel specification Forth : a minimalist kernel specification that leaves it up to the implementor to expand the scope. He pointed me to *'Thinking Forth: A Language and Philosophy for Solving Problems'* by Leo Brodie. "The smallest atom of a Forth program is not a module or a subroutine or a procedure, but a 'word'. Furthermore, there are no subroutines, main programs, utilities, or executives, each of which must be invoked differently. Everything in Forth is a word." (Brodie, 2004, p. 19)

5.3.7.2 Cartesian Product Topology (DWDS Lookup Tables)

A Cartesian product table topology is employed where the purpose is to obtain a correct value from a structured data set. The designer would use available (GIVEN) context data and available (WHEN) particulars data to swiftly pick out a discrete value (THEN). The DWDS employs the Cartesian product topology for all its 'lookup tables' with the data configured into ordered lists. Field names are at the head of each column, and cells below containing the values. This is referred to as a 'tuple':

customer_id	**firstname**	**lastname**
758347	Antonio	Remitz
382639	Jessie	Lee
937465	Paula	Gaviria
828682	Amy	Green
673837	Walter	Staniszewski

Such reference data is stored and processed in JSON form, as follows:

```
[
{ "customer_id" : 758347, "firstname" : Antonio, "lastname" : Remitz },
{ "customer_id" : 382639, "firstname" : Jessie, "lastname" : Lee },
{ "customer_id" : 937465, "firstname" : Paula, "lastname" : Gaviria },
{ "customer_id" : 828682, "firstname" : Amy, "lastname" : Green },
{ "customer_id" : 673837, "firstname" : Walter, "lastname" : Staniszewski }
]
```

Three lookups 'required' for every DWDS request-response operation are:

- ISO Codes for the representation of names of countries and their subdivisions (ISO, 2020)
- The International Standard Industrial Classification (UNSD, 2018)
- The United Nations Standard Products and Services Classification (UNDP, 2018)

The schema for the row labels in a Cartesian product table must be 'canonical', meaning that it has been semantically refined so that there is a unique name (in each implemented language) for each category of its domain, and it maintains consistent technical attributes for the data in the cells.

Table 20 illustrates a two-level canonical schema for rows of a Cartesian product table for a [lookup.dwd] record containing seven data inputs, two of type {i, ii}, two of type {p, q} and three of type {a, b, c}, resulting in 2 × 2 × 3 = 12 data output possibilities. Various cell entries such as "b,ii,q" represent target THEN data at a second level of WHEN conditions. The same data is shown for comparison using the vertical topology in Table 21. DWDS [lookup.dwd] tables use the Cartesian Product topology, whereas [rule.dwd] uses the Vertical Stack arrangement, but as can be seen with this comparison, [lookup.dwd] and [rule.dwd] tables are not very different in their underlying data structure. Furthermore, from Table 21 it is easy to imagine storing this data in a single row of the RuleReserve via the Horizontal Tape topology. Just line up the data from Table 21 row-by-row, end-to-end, from the first row to the last (see Section 5.3.7.4).

GIVEN			WHEN a	WHEN b	WHEN c
	WHEN i				
		WHEN i,p	THEN a,i,p	THEN b,i,p	THEN c,i,p
		WHEN i,q	THEN a,i,q	THEN b,i,q	THEN c,i,q
	WHEN ii				
		WHEN ii,p	THEN a,ii,p	THEN b,ii,p	THEN c,ii,p
		WHEN ii,q	THEN a,ii,q	THEN b,ii,q	THEN c,ii,q

Table 20: A generic Cartesian Product DWDS Table with 7 input conditions

Table 21: A Vertical Stack representation of the data from Table 20

GIVEN	Sentence	A	B	C	D	E	F	G	H	I	J	K	L
	WHEN a	1	1	0	0	0	0	1	1	0	0	0	0
	WHEN b	0	0	1	1	0	0	0	0	1	1	0	0
	WHEN c	0	0	0	0	1	1	0	0	0	0	1	1
	WHEN i	1	1	1	1	1	1	0	0	0	0	0	0
	WHEN ii	0	0	0	0	0	0	1	1	1	1	1	1
	WHEN p	1	0	1	0	1	0	1	0	1	0	1	0
	WHEN q	0	1	0	1	0	1	0	1	0	1	0	1
	THEN a,i,p	1	0	0	0	0	0	0	0	0	0	0	0
	THEN a,i,q	0	1	0	0	0	0	0	0	0	0	0	0
	THEN b,i,p	0	0	1	0	0	0	0	0	0	0	0	0
	THEN b,i,q	0	0	0	1	0	0	0	0	0	0	0	0
	THEN c,i,p	0	0	0	0	1	0	0	0	0	0	0	0
	THEN c,ii,p	0	0	0	0	0	1	0	0	0	0	0	0
	THEN a,ii,p	0	0	0	0	0	0	1	0	0	0	0	0
	THEN a,ii,q	0	0	0	0	0	0	0	1	0	0	0	0
	THEN b,ii,p	0	0	0	0	0	0	0	0	1	0	0	0
	THEN b,ii,q	0	0	0	0	0	0	0	0	0	1	0	0
	THEN c,ii,p	0	0	0	0	0	0	0	0	0	0	1	0
	THEN c,ii,q	0	0	0	0	0	0	0	0	0	0	0	1

If there is a [lookup.dwd] assigned the identifier 4d4673b6-3f5d-499b-a87e-468bd0b2c268, then one of the Input Conditions of a [rule.dwd] record could say: "The [daily_indexed_price] of the delivery service is the [framework_base_price]*[b,ii,q]:4d4673b6-3f5d-499b-a87e-468bd0b2c268". This contains enough information, and the DWDS platform would embody the necessary procedures for a fast simple lookup to the "b,ii,q" cell in the targeted [lookup.dwd] table. By this method, forward and backward chaining among [rule.dwd] and [lookup.dwd] records is exceedingly simple.

5.3.7.3 Vertical Stack Topology (DWDS Logic Gates)

Input/output tables are commonly used for representing logical relations, with cells constrained to binary or trinary symbols like {0,1} for purely operational logic, or {T,F} for epistemic logic. This makes the reasoning structure easy for humans and computers to perceive and process, even when the relationships are fairly complicated, because it strips bare the logic relations of each rule into a simple array of symbols within the cells, and shifts the semantic weight to the labels of each row and column.

The vertically stacked table topology comprises a type of virtual logic gate for selecting one or more scenarios described by two or more statements. The designer's purpose is to use available (GIVEN) context data, and available (WHEN) particulars data, to identify the intended set of output statements (THEN).

This table structure requires that the labels for the rows be 'non-canonical' because more than one row is needed to distinguish at least two different columns. This table topology is typically employed with binary or trinary sets of symbol, but higher-dimension sets such as tetranary symbol sets work as easily.

5.3.7.4 Horizontal Tape Topology (DWDS 'RuleReserve' Tables)

The DWDS uses a horizontal tape topology for storage and retrieval with a large distributed table of [rule.dwd] and [lookup.dwd] records. This is essentially a virtual telex tape variant of Alan Turing's horizontally-splayed table, which he first described as follows:

> "Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book. In elementary arithmetic the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, i.e. on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite. (Turing, 1937, p. 249)"

Current-generation 'wide-column' data fabrics employ this topology for distributed storage and querying of structured data (Chang et al., 2008) (Carpenter & Hewitt, 2016) (Mahgoub et al., 2017). These are virtual manifestations of an unlimited number of physical Turing tapes laid out side-to-side. We can imagine that each is punched like a telex strip with short vertical sets of 8 squares that have or do not have a punched hole. These permutations of holes and blanks represent binary numbers between the value 0000 0000 (all blanks) and the value 1111 1111 (all holes), that's to say between 0 and 255. Data in this form can express meaning through sequences of the 255 available *symbols* to create semantic *signals* in any language. One may then place end-to-end on the horizontal tape each row of any two-dimensional logic table, or each row of any reference table. Any number of these virtual tapes can then be laid side-to-side to create one enormous consolidated [*m* x *n*] matrix (i.e. *rows* x *columns*), such that the tape constitutes an indexed row. This is not easy to illustrate, but that's to be expected because this topology is optimized for machine storage efficiency and processing speed. Table 22 portrays a segment of such a table. This layout retains auditability but a specialized search algorithm would be needed to make this practical.

Table 22: DWDS [rulereserve.dwd], [is.dwd] and [ought1.dwd] Tables

Horizontal tape topology (single row per rule) for optimal messaging, storage and retrieval

Context	Statement	A	B	C	D	E	F	...	Statement	A	B	C	D	E	F	...	Statement	A	B	C
GIVEN	WHEN 1.1	1	1	0	0	0	0	...	WHEN 1.2	0	0	1	1	0	0	...	WHEN 1.3	0	0	0

This *Horizontal Tape Topology* enables compact storage of large numbers of rules, one rule-per-row, all expressed with the DWDS RuleData, as well as extremely fast discovery and high-volume processing. *Information* was defined earlier as data with context. Rules-as-data in this configuration is optimized for machines and networks, while the other two topologies discussed above trade off some machine performance for human readability. The [rule.dwd] and [lookup.dwd] records are stored in a horizontal [rulereserve.dwd] table, across a decentralized, distributed network of RuleReserve (RR) nodes, one row per record. Each [is.dwd] message is also pre-formatted horizontally, so that it can be used directly as a sieve, with no need for reconfiguration during run-time. As each [is.dwd] message arrives to RuleReserve or RuleTaker in this horizontal tape form, it is ready-to-go as an *in-memory* seive in a fully 'expected' form. With no transformation delays the data sifting operation can proceed instantly. From an information thermodynamics perspective, one can imagine that RuleReserve and RuleTaker are like data batteries charged with potential exergy. Whenever an [is.dwd] filter arrives to distinguish a signal from the noise, they release a burst of information. This is done with very primitive *signal*-matching. An [is.dwd] message contains (GIVEN) context data, which is used by an RR node to sift out all the rows from a consolidated [*m* x *n*] matrix of DWDS structured data which contain the signals of rules that are 'in effect' for that jurisdiction,time/date, and identity. Once sifting for context is completed, the (WHEN) particulars data from the same [is.dwd] message is used to sift for rows that contain signals that are 'applicable'.

The resulting subset of the initial [*m* x *n*] matrix constitutes the GIVEN-WHEN-THEN information required to assemble an [ought1.dwd] record, which nevertheless is acknowledged to have several limitations:

- It can never be presumed that the supplier of the [is.dwd] message data has provided a complete description of the particular circumstances. An incomplete [sieve1.dwd] can miss some 'rules'.
- It can never be presumed that the DWDS RuleReserve contains the full set of the rules 'in effect' for a jurisdiction. The work of creating and maintaining [rulereserve.dwd] is collaborative and ongoing.
- The prerogative for genuine 'execution' of the rules is necessarily external to this service.

5.3.8 Externalize Reusable Algorithms (In-Memory Retrieval of Cartesian Product Tables)

Cognitive psychologists Pierre Barrouillet and Michel Fayol have documented a set of factors which lead to a shift in human thought processes from procedural calculation to "faster and less costly memory retrieval of the items of information" within the human brain. This involves "moving from an algorithmic strategy ... toward a strategy for the direct retrieval of results from memory". (Barrouillet & Fayol, 1998, 364-66)

A similar shift to fast in-memory retrieval can also be employed at scale with computers. But this method is so obvious that it is commonly overlooked. We'll illustrate this with simple table of pre-calculated data that can be used to externalize the work of an algorithm from the run-time burden.[56]

Suppose a project manager needs to estimate the transportation requirements for shipping square timber that will be produced from a woodlot, in which the number and average size of trees is known. Let's also say that the capacity of available flatbed trucks is known. The manager needs to estimate the amount of square timber of the available round logs, in cubic feet. This is just one of many estimates that would need to be done throughout the logistical supply chain; there are also contract and compensation estimates, fiscal fees, cargo planning, engineering load safety assurance, budgeting, and other purposes.

56 The philosopher Ludwig Wittgenstein and the psychologist Wolfgang Köhler both examined "ways the obvious fails to be obvious and the roles that *describing aspects of ordinary, common life* play in lifting one's blindness to the obvious" (Dinishak, 2014, p. 61, emphasis added)(Dinishak, 2014, p. 61, emphasis added)(Dinishak, 2014, p. 61, emphasis added).

But to estimate the amount of square timber that can be produced from the available round logs, someone could write an application that would include various formulas and estimation rules to be run with each query. But this can also be written as a simple lookup to an informatic rendering of the 100-year-old reference table shown in Table 23:

- GIVEN the industry is forestry
- WHEN the round log length is 24 feet, and the round log average diameter is 24½ inches
- THEN the square timber volume is 45 cubic feet

Table 23: Example of Pre-Processed Tabular Data in a Cartesian Product Table (Taylor, 1904, Reprinted 1951)

Contents of Round Logs

CUBIC CONTENTS IN SQUARE TIMBER OF ROUND LOGS AS CUT FROM THE TREE

Feet Length	Log 22 in. Thick	Log 22½ in. Thick	Log 23 in. Thick	Log 23½ in. Thick	Log 24 in. Thick	Log 24½ in. Thick	Log 25 in. Thick	Log 25½ in. Thick	Log 26 in. Thick	Log 26½ in. Thick
	cu. ft.	cu. ft.	cu. ft.	cu. ft.	cu. ft.	cu. ft.	cu. ft.	cu. ft.	cu. ft.	cu. ft.
20	30	31	33	34	36	37	39	40	42	43
21	32	33	34	36	38	39	41	42	44	46
22	33	34	36	38	39	41	43	44	46	48
23	35	36	38	39	41	43	45	46	48	50
24	36	38	39	41	43	45	46	48	50	52
25	38	39	41	43	45	46	48	50	52	54
26	39	41	43	44	46	48	50	52	54	56
27	41	42	44	46	48	50	52	54	57	59
28	43	44	46	48	50	52	54	56	59	61
29	44	45	48	50	52	54	56	58	61	63
30	45	47	49	51	54	56	58	60	63	65
31	47	49	51	53	55	58	60	62	65	67
32	48	50	52	55	57	59	62	64	67	69
33	49	52	54	56	59	61	64	66	69	72
34	51	53	56	58	61	63	66	68	71	74
35	52	55	57	60	62	65	68	70	73	76
36	54	56	59	61	64	67	70	72	75	78
37	55	58	61	63	66	69	72	74	77	80
38	57	59	62	65	68	71	73	76	79	82
39	58	61	64	67	70	72	75	78	82	85
40	60	63	65	68	71	74	77	80	84	87
41	61	64	67	70	73	76	79	82	86	89
42	63	66	69	72	75	78	81	84	88	91
43	64	67	70	73	77	80	83	86	90	93
44	66	69	72	75	78	82	85	88	92	95
45	67	70	74	77	80	83	87	90	94	98
46	69	72	75	79	82	85	89	92	96	100
47	70	74	77	80	84	87	91	94	98	102
48	72	75	79	82	86	89	93	96	100	104
49	73	77	80	84	87	91	95	98	102	106
50	75	78	82	85	89	93	97	100	104	108

Shown here is one page from a six-page table available in the “Ready Reckoner” pocket-book originally published in 1904 (from the 1951 reprint). The full table provides pre-calculated results for round log lengths ranging from 12 to 42 feet, that are from 10 to 40 inch diameters, in half-inch increments. Following is the explanation of how the results were derived to create the table:

> *"To reckon the contents of a round log in cubic feet of square timber... Measure the diameter at each end in inches; add those measurements together, and divide the sum-total by 2; the quotient is the average diameter. One third of this diameter is allowed for the chips or slabs. To deduct this third, divide the number of inches diameter by 3 and subtract the quotient from it; the remainder is the proper diameter for measurement. The thickness of the log is generally counted in even inches, and one-third of an inch excess, or upward, is added as an extra inch. After getting the square of the log in the manner above described, the number of cubic feet in it is reckoned the same as the square timber. ... Measure the thickness of the stick each way in inches and multiply one by the other, then multiply the product by the length."* (Taylor, 1951 p. 226, 237)

All the relevant answers are pre-calculated in this table. Writing the formulas for this method would not be technically difficult, but would be unnecessarily convoluted in any programming language compared with an instantaneous lookup to such a table if it were freely available on the Internet in a generic format, and replicated locally to reduce latency. Such a table lookup action would be almost as fast as a ping. This low-tech method can speed up data processing in the same way that most people recall by rote memory that "three fives are fifteen" without actually performing the arithmetic operation in their heads (Barrouillet & Fayol, 1998). The data processing required to work out the answer is thus externalized *once and for all*, literally. When this table is available online, any application in any programming language, operated by anyone for any purpose, could readily perform a lookup to get the answer, keeping their computing resources focused upon whatever primary task they may have.

5.3.9 Externalize Declarative Conditions and Assertions from Logical Relations

5.3.9.1 Roles and Purposes

In order to obtain a consistent structuring of rules which would be straightforward for humans to comprehend, and efficient for computers to process, the DWDS separates the sets of declarative sentences which express the input conditions and output assertions of rules, as well as from their logical relations and normative modes (MUST, MAY, SHOULD). Logic and mode are placed into an adjacent array, in such a way that each rule can be efficiently stored, discovered and transmitted as data, for display to a human through any device interface, and for fast machine processing on non-specialized systems. This method side-steps a great deal of complexity in rule expression by externalizing it to human or machine agents. Complicated rules found in source documents can be expressed using simple structured statements and numeric signals. This leaves rule sifting and processing as simple and deterministic.

The first step towards the DWDS logic structure involved experimentally adapting the Decision Model and Notation (DMN) standard (OMG, 2019a) adapted to accommodate complete natural language sentences in the column headings, as shown in Table 24 as an intermediate comfiguration. Others, such as Marjolein Deryck (Deryck et al., 2019) have also adapted the standard DMN layout for particular requirements. Our present operational technique has involved creating a vertical I/O tabular logic gate comprised of trinary input (true-false-both) and binary output (true, false), with full, but uniformly structured, normative statements for semantics.

Table 24: DMN Table Adapted as an Asymmetric Trinary|Binary Logic Gate

Grocery Store Delivery Policy							
Unique Rule ID	R4qG9UeKH5hW	**INPUT.CONDITIONS**			**OUTPUT.ASSERTIONS**		Annotations
Declarative Expression Components	The [Verb, Past Participle]	The used	The measured	The contained	The described	The proposed	*A reported state of*
	[Attribute] of the	capacity of	type of	value of	grocery	delivery service	*a particular attribute of*
	[Noun for Subject]	this box	this box	this box	delivery service	fee	*this subject*
	[Verb, Singular Present]	is	is	is	is	is	*requires testing against*
	[Adjective OR arithmetic/boolean	>=0.5	standard	>=$100	offered	>$0.00	*this requirement.*
Scenarios	A	F	B	B	F	F	Delivery is available
	B	T	B	F	T	T	Delivery is charged
	C	T	T	T	T	F	Delivery is free
Annotations	hit policy: unique match only	as reported by	Lhw 51.0cmx3	matching the pre	to approx. 10 km	In accounting re	Xalgo Augmented DMN Table v0.1.2

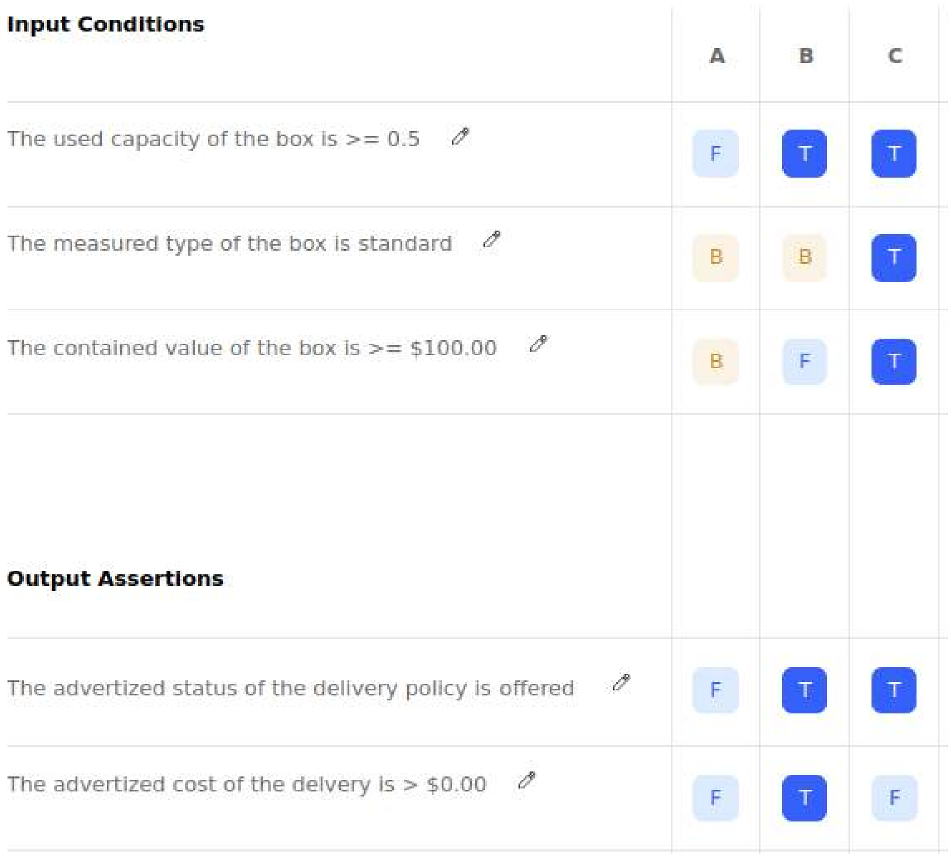


Figure 16: A Graphical Asymmetric Trinary|Binary {F, T, B | F, T} Logic Gate

Though the design process we arrived at a simpler way to show the same information, as in Figure 17. (Note that tables containing graphics are referred to here as 'figures'.) Some variations discussed later in this dissertation are previewed in Figure 18 with numeric elements that have no intrinsic semantic meaning but are convenient for keyboard expression and functional data processing, and Figure 19 using symbols with explicit semantic meaning to humans, but which are not suited to keyboard expression or functional data processing.

This emerging compartmentalized structure offers several practical advantages:

- *Rule-Maker Agent*

  Anyone composing or editing a rule can more easily perceive, understand and associate each declarative statement on its own as a discrete normative proposition, so that;

  - The author of a rule can readily discern the input permutations that lead to unique output permutations. This improves clarity and reduces errors.
  - The author of a rule can more readily notice when different input permutations lead to the same output permutations, so that these can be consolidated into a single column. Consolidation of scenarios with common outputs improves data processing performance, and decreases file size.

- *Rule-Taker Agent*

  Human comprehensibility and machine computability of the various conditions and assertions of a rule are facilitated by laying out side-by-side the normative proposition, the normative modes and their logical relations.

  - Cause and effect within a rule becomes simple for any person to perceive;
  - The compartmentalized data can be used directly by any application running any programming language or database.

- *RuleReserve*

  Stored normative propositions can be stored efficiently and processed quickly.

  - The array of logic relations and normative data is easily transposed into a horizontal tape topology (row) for compact auditable storage, and extremely fast in-memory processing.
  - Removing the logic relation and the normative mode from the normative proposition sentences facilitates lightweight symbol-matching, lookups and arithmetic/Boolean operations.

Input Conditions		Scenarios		
		A	B	C
1	The used capacity of the box is ≥ half full	00	01	01
2	The measured type of box = standard	11	11	01
3	The contained value of the box is ≥ $100.00	11	00	01
Output Assertion				
i	The described status of the delivery service = offered	00	01	01
ii	The advertized price of delivery ≥ $0.00	00	01	00

Figure 17: A Graphical Asymmetric Trinary|Binary Logic Gate

<u>Above</u>: Numeric for keyboard typing and fast processing {00,01,11 | 00, 01}

<u>Below</u>: Symbolic for easy human interpretation {X, √, ? | ⊘, !! }

Input Conditions		Scenarios		
		A	B	C
1	The used capacity of the box is ≥ half full	X	✓	✓
2	The measured type of box = standard	?	?	✓
3	The contained value of the box is ≥ $100.00	?	X	✓
Output Assertion				
i	The described status of the delivery service = offered	⊘	!!	!!
ii	The advertized price of delivery ≥ $0.00	⊘	!!	⊘

This style of logic table borrows from Alonzo Church's framing of Gottlob Frege's distinction between sense and reference (Frege, 1892) by employing declarative 'assertions' as 'names'. (Church, 1956, p. 23-24) In our design, each Input Condition and Output Assertion sentence is the name (label) for a row in an array of data that specifies the logical relations and normative modes. The subject-predicate-object triple within each whole declarative sentence describes a non-canonical (unique) state, which functions as the label for a row in this tabular logic gate, so that permutations of input signals can be decisively mapped to permutations of output signals. The logical relations are arranged by rule authors assigning symbols {00,01,10,11} or {X,√,&,?} to each of two or more input declarative statements, and then assigning symbols {00,01,10,11} or { ⃠,!!,○,! }. Permutations of Input Conditions ('IS') are associated with ('RULE') the various Output Assertions ('OUGHT'), to express IS + RULE ⟹ OUGHT.

5.3.9.2 The DWDS Normative Logic Gate

The '*DWDS normative logic gate*' is a tetranary structure that adapts the Damjanović-Sirakoulis {0,1,2,3} numeric data model into a vertical I/O table structure. (Hereinafter the phrase "vertical I/O" table or logic gate refers to the layout illustrated in Figure 17 and following.) The novel semantic assignment is illustrated in Figure 18. Normative logic is not be premised upon any single or timeless truth; rather it provides the parties to an interaction with a method for contingent agreement upon what they each consider to be 'in effect', 'applicable' and 'invoked', qualified by their degree of mutual trust.[57] Anyone who prefers the long-established "truth table" convention may configure their implementation to display the letters {T,F}, but in the DWDS data system, these can just as well be taken to signify T ≡ *token* and F ≡ *foo*. The numeric tokens {00, 01} are more directly suited to the naïve empiricism and intrinsic multilingual expressiveness of our intended design. They are free of any external epistemological implication. Computationally, they are just markers.

The color scheme is thoughtfully considered: grey (dbdee6), navy (081a59), purple (a523f0), and orange (f06318). These are intended to support easy visual parsing for diverse eye types (Brewer, 2013), and to avoid potential confusion with global standard green-amber-red semantics, while also reflecting some meaning relevant to logic gate operation:

57 Earlier we highlighted Carlos Alchourrón's argument that declarative propositions cannot be assumed to be complete, accurate or consistent, which led to Georg von Wright's concession that "experience seems to testify that mutually contradictory norms may co-exist within one and the same legal order, and also that there are a good many 'gaps' in any such order" (Von Wright, 1999, p. 20)

- {00,01} are unambiguous, therefore they are represented with simple grey and black (navy):
  - 00—Neutral grey implies input NO and output NOT (absence);
  - 01—The dark navy brings visual attention to the input YES and the output MUST;
- {10,11} are ambiguous, therefore the chosen colours alert the viewer to "pay attention":
  - 10—Purple is an international informal standard color for *[radio]active*. Our use here for input YES AND NO, and output MAY expresses ambiguity to elicit user reaction;
  - 11—Orange is an international formal standard color for alerts. Our use here for input YES OR NO, and output SHOULD is intended to elicit user attention.

After experimenting with various colors, it felt easy to cognitively filter in order to 'look at' only the grey and navy parts of this logic gate when desired. The brightness of the colors purple and orange let the viewer passively notice and distinguish between them.

Figure 18: Semantic Assignment of Numeric Tetranary Elements for DWDS Logic Gate States

Binary	Symbol	Input Conditions	Output Assertions
00	00	NO	NOT
01	01	YES	MUST
10	10	YES AND NO (BOTH)	MAY
11	11	YES OR NO (UNSURE)	SHOULD

During review with various colleagues, this purely numeric representation requiring separate meanings in the input and output sections of the logic gate, was felt to be sub-optimal because it requires the reader to parse them differently. So I put forward a set of (extended) ASCII symbols: INPUT {X,√,&,?} and OUTPUT { ⊘, !!,○,! }.

The full descriptive name of our scheme in the style of the earlier list is thus:

- {00,01,10,11 | 00,01,10,11} — DWDS Symmetric Tetranary Vertical I/O Logic Gate (Numeric)
- {X,√,&,? | ⊘,!!,○,! } — DWDS Symmetric Tetranary Vertical I/O Logic Gate (Symbolic)

Both the numeric and symbolic variants are included as user interface options of DWDS RuleMaker. However the stored and transmitted data is only numeric. Also, throughout this dissertation we use the numeric elements because √, ⊘ and ○ are not keyboard characters.

Figure 19: Semantic Assignment of Symbolic Tetranary Elements for DWDS Logic Gates

Binary	Input Symbol	Input Conditions	Output Symbol	Output Assertions
00	X	NO	⦸	NOT
01	✓	YES	!!	MUST
10	&	YES AND NO	O	MAY
11	?	YES OR NO	!	SHOULD

When a rule author wants to accommodate uncertainty directly into a rule, this is expressed in the Input Conditions of the vertical I/O table with the value 10 meaning Yes-AND-No or 11 for Yes-OR-No. A rule author can also avoid unwarranted rigidity in their Output Assertions for nuanced human relations with 10 for MAY and 11 meaning SHOULD. The DWDS thus extends well beyond, for example, the `uncertaintyType` tag that is part of the HL7 standard (HL7, 2019a) (HL7, 2019b).

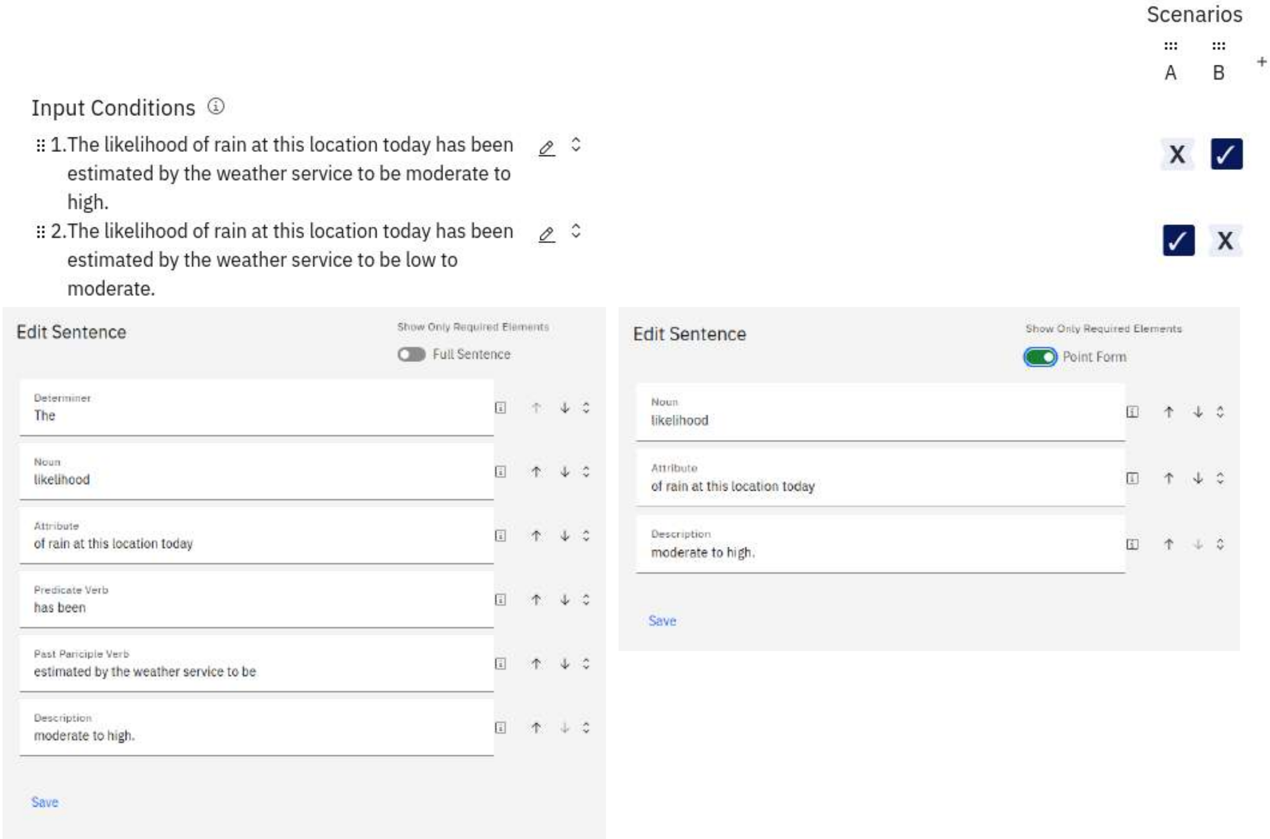
The DWDS logic gate data model is intended for Internet-wide usage with all rule types, on all platforms. This requires that it be simpler and more generalizable than other currently available methods of logic expression for data processing.[58] A rule author should only need to supply the relevant Input Condition sentences to adapt the DWDS logic gate to achieve any particular logic framework. Although the DWDS logic gate is structured with discrete sentences, there are several ways to accommodate continuous variables among the Input Conditions. Following are three hypothetical illustrations: chunking of continuous data via qualitative terms in Figure 20, chunking of continuous data via estimation in Figure 21, and chunking of continuous data via ranges in Figure 22. The first of these three includes the edit boxes of the RuleMaker interface to demonstrate sentence construction with the six syntactic elements.

58 The potential for adding the Boolean logic gates NOT, NAND, NOR, XOR, and XNOR directly into RuleMaker was considered and experimented with for the DWDS design. However it was found that any of these possibilities can be handled with additional sentences and scenarios -- e.g. a sentence for MUST, and a sentence for MUST NOT. Adding the capability via double icons or additional icons would reduce comprehensibility for the majority of users, and complicate the overall DWDS data model and the designs for RuleMaker and RuleTaker, merely to obtain results that can already be expressed with some thoughtfulness via the simpler system. Personally I found that I had to do more thinking in how to use NAND and XNOR correctly, than I require to achieve the same result currently.

- *Chunking of Continuous Data with Qualitative Terminology (Figure 20)*

An outdoor event company implements a rule that would trigger a message suggesting to ticket-holders that due to the chance of rain, they should bring raincoats or umbrellas. The input conditions can be loosely worded, relying on keywords at the website of the weather service.

Figure 20: Chunking of continuous data with qualitative terms in a DWDS logic gate



- *Direct Chunking of Continuous Data with Estimation (Figure 21)*

Suppose respondents are asked to indicate their confidence in a proposal by selecting one of the five levels: 0%, 25%, 50%, 75%, and 100%. A functional rule could be that *when a respondent expresses anything other than full confidence as 0% or 100%, then more information is required.*

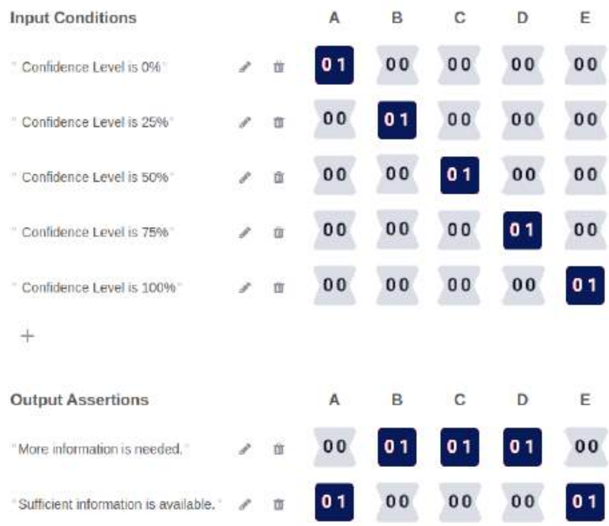


Figure 21: Direct chunking of continuous data with estimation in a DWDS logic gate

- *Chunking of continuous data with ranges (Figure 22)*

Continuous gradient source data can be handled in a DWDS logic gate through the use of some simple opportunistic techniques that avoid grouping the data itself, even while applying a discrete multi-value technique. Two techniques are combined here: (a) use of a derivative of the raw data; and (b) use of the Yes-AND-No element so that any momentary or temporary occurrence within a time period would be sufficient to satisfy the condition. Let us imagine, then, a hypothetical forensic analysis of a mishap at a construction site that seems related to wind gusts, in which due diligence for risk management is being assessed with rules that are applicable after-the-fact. Continuous wind speed data from a nearby anemometer is assumed to be available for the hour prior to the event. A standard or guideline used by this investigation team could state: *"Basic precautions are considered optional so long as wind gusts (the first derivative of wind speed, i.e. 'acceleration') remained less then 0.5 mps^2 (meters per second squared). Basic precautions would be* *expected where there were any wind gusts within that hour which were greater than 0.5 mps² but less than 1.5 mps². And enhanced precautions would be deemed mandatory if there were any wind gusts in that hour between 1.5 mps² and 2.0 mps².”*

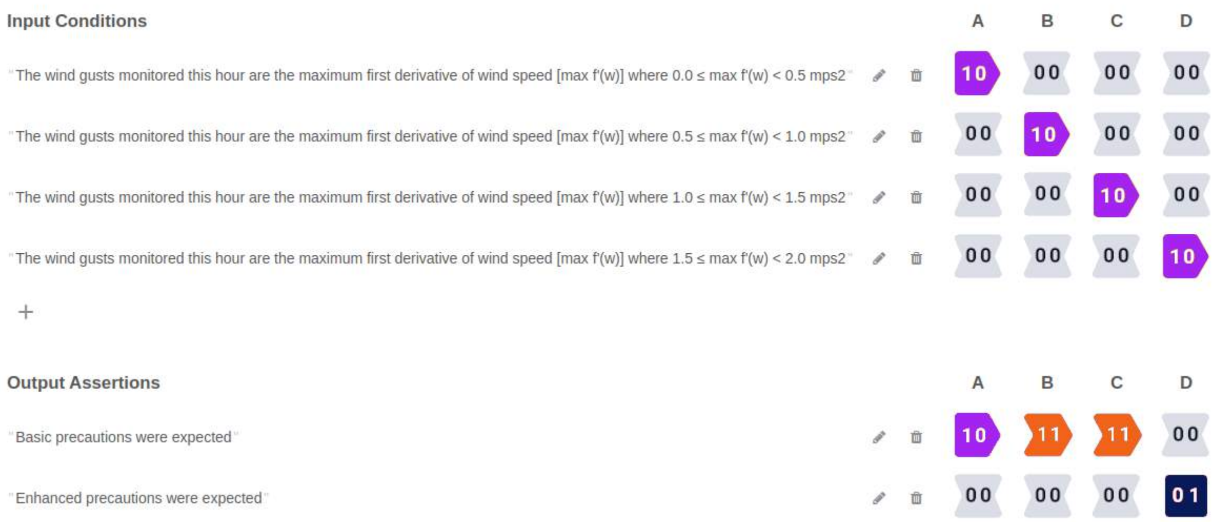


Figure 22: Indirect chunking of continuous data with ranges in a DWDS logic gate

5.3.9.3 The ‘DWDS Logic Gate’ Differs from a ‘Decision Table’

Whereas the purpose of a decision-table is to arrive to a single unambiguous determination the with available information, the purpose of a DWDS logic gate is only to inform agents, human or machine, about rules which are 'in effect' (prerogatives and time) and 'applicable' (categories), and 'invoked' (for a particular circumstance), in such a way that *they are more informed while making their decision*. The DWDS logic gate can potentially have more than one outcome when the Yes-AND-No or Yes-OR-No elements are in use, unlike a ‘decision table’ which has only one outcome.[59] The DWDS logic gate accommodates scenarios where the decision is understood to be the prerogative of, or is delegated to, a human end user. (Vanthienen, personal communication, April 13, 2022)

It could well be that the data model and logic structure of a rule ought to be refined in order for a given rule to arrive at a single determination. But neither the end user nor the designer of a rule system typically possess the prerogative to rewrite rules or software applications of others. The DWDS specification is designed to operate with rules as they are promulgated by imperfect people and institutions, and to accommodate available information about the state of the world, constrained

59 This distinction was emphasized by Jan Vanthienen, the preeminent historian of decision table methods, and professor of information systems at KU Leuven (Belgium), Department of Decision Sciences and Information Management, during a conversation in the second week of April 2022.

as it is. The real world is characterized by sub-optimal rules, applications, databases, communications and data. A DWDS implementation of an Internet of Rules would be of no utility if it depended upon first making all the uploaded laws, policies and contracts gapless and contradiction-free. On the other hand, DWDS implementation does help to make gaps and contradictions more evident, and it builds the human agent's role into the resolution of uncertainty so problems can be recognized and fixed.

It is useful to distinguish among distinct sources of uncertainty, such as: (a) lack of precision in terms used to express the input conditions of rules; versus (b) lack of confidence in the input data that would invoke the conditions of rules. The organizational theorist Martin Tribus once emphasized "the difference between Mr. A, who is uncertain about knowing at which airport his plane will land, and Mr. B, who is uncertain about everything. To be uncertain about everything represents a lesser state of knowledge than to be uncertain about something definite." (Tribus et al., 1966, p. 245)

5.3.9.4 Discussion: 'DWDS Logic Elements' Differs from the 'Wright / Ostrom School'

For four days, 19-22 April 2022, I was one of a dozen invited participants in the Rules-as-Data workshop hosted by the European Consortium for Political Research. (Damonte & Bazzan, 2022) This was co-chaired by Alessia Damonte, senior lecturer in political science, Università degli Studi di Milano, and Giulia Bazzan[60], postdoctoral researcher, University of Copenhagen. During a break, I had an opportunity to discuss the normative scope of my DWDS design with Christopher Frantz and Saba Siddiki.[61] Their "Institutional Grammar 2.0" specification is based upon the "Grammar of Institutions" produced a quarter century ago by Nobel Laureate Elinor Ostrom (Crawford & Ostrom, 1995) (Ostrom & Crawford, 2005).

I share Ostrom's rationale on common pool resources (Ostrom, 1990), but in the course of my own research and reflection on this topic, I decided against adopting the Ostrom-Crawford structure of deontic logic. My reasons concern the formulation of the Input Conditions and Output Assertions:

60 Author of: *Effective Governance Designs of Food Safety Regulation in the EU: Do Rules Make the Difference?* (Bazzan, 2021)

61 Co-authors of the widely-used "Institutional Grammar 2.0: A Specification for Encoding and Analyzing Institutional Design", and of the new book published in 2022 by Springer: *Institutional Grammar: Foundations and Applications for Institutional Analysis*. (C. K. Frantz & Siddiki, 2022)

(a) *DWDS 'Input Condition' States Differ from the States in Ostrom's Logic Scheme* Elinor Ostrom built upon the deontic structure of 'early career' Georg von Wright's gapless and contradiction-free (determinate) *systems of norms* in the 1950s (Von Wright, 1951). But by the 1970s, 'later career' von Wright had accommodated the view of Carlos Alchourrón and Eugenio Bulygin, to acknowledge that gaps and contradictions (indeterminacy) characterize real *systems of norm propositions*. The DWD Specification augments *{No; Yes}* to incorporate uncertainty though [10] Y*es-AND-No and [11] Yes-OR-No.* The path by which I arrived at this tetranary logic structure in the course of my own design research on the DWDS was via tetralemma logic (Gunaratne, 1980) (Priest, 2010) (Madanayake et al., 2015), after a suggestion about it from Wayne Cunneyworth (Cunneyworth, 1994) (Cunneyworth & Olders, personal communication, June 4, 2020). Although this would seem to make the method more abstract, I find its flexibility actually makes DWDS more suitable to genuine rule samples and scenarios from the real world.

(b) DWDS 'Output Assertions' Differ from the Options in Ostrom's Logic Scheme

The scope of output assertions in Ostrom's deontic logic are also drawn from von Wright's 1950s scheme involving 'may' (permitted), 'must' (obliged), and 'must not' (forbidden) (Von Wright, 1951). To these she combines SHOULD into MUST, distinguishing SHOULD only as a low-intensity MUST:

> "Generally, in everyday language, "must" obligates someone more strongly than "should," and "must not" forbids someone more strongly than "should not." ...[D]elta parameters allow more precision in the weight of the Oblige or Forbid and thus can be used to distinguish between "should" and "must" when needed in analysis. (Ostrom & Crawford, 2005, p. 142) (Crawford & Ostrom, 1995, p. 587)

In my view the Wright/Ostrom school overlooks the most essential difference between MUST and SHOULD: the social relation between the rule-maker and the rule-taker, namely:

- 'MUST' prioritizes *rule-maker prerogative* in the social relation; but
- 'SHOULD' prioritizes *rule-taker prerogative* in the social relation.

I also suggest that at the operational level a single parameter cannot achieve the "precision in the weight ... to distinguish between 'should' and 'must'" if the intent is to mimic what natural languages achieve with a nuanced range from gentle suggestion to strong exhortation. For this purpose, DWDS instead provides rule-makers and rule-takers a method to append three sliding scale criteria for sorting, adapted from the work of linguists An Verhulst, Ilse Depraetere and Liesbet Heyvaert (Verhulst et al., 2013):

(a) *Source:* de jure authority and/or de facto origins;

(b) *Subjectivity:* commitment of beneficiaries towards ensuring fulfillment; and

(c) *Strength:* gravity or the impossibility of non-compliance.

For these two reasons my DWDS design builds upon industry standards (Bradner, 1997) (ISO/IEC, 2018) which distinguish three primary semantic elements MUST/SHALL, MAY/MIGHT, SHOULD/OUGHT TO. This includes their negatives: obligation-obstruction; permission-prohibition; encouragement-discouragement.

5.3.9.5 Discussion: 'DWDS Logic' is Combined with Metadata to Find and Fetch

The DWDS design provides to any generally-competent person in business, commerce, industry or government a readily accessible means to transcribe their unstructured rules from legislation or contracts into simple controlled natural language sentences, and from there, automatically into machine-parsable, platform-agnostic JSON, published to the Internet. But they are only useful on the Internet if they can be found and fetched instantaneously on-demand.

None of the other currently available rule systems reviewed during this research addressed the requirement to find and fetch rules that the rule-taker (including an application designer) does not yet know about. They do not address the requirement of rule-makers to ensure on-demand delivery of their rules to all parties who ought to know about them. This is a unique contribution of the DWDS design for the decentralized Internet.

During the Internet Engineering Task Force (IETF) Meeting 105 in 2019, I scheduled a side meeting with Henrik Levkowetz, Chair of the IETF Tools Team. That discussion led to my subsequent draft proposal to the IETF (General Area; RFC-Editor) entitled *"An Experiment in Automated Discovery of 'In Effect' and 'Applicable' Technical Standards Requirements During Unit Testing"*. This proposal has not *yet* been acted upon because (a) my DWDS design was not completed until December 2021; (b) the reference implementation of DWDS software was not ready until spring 2022; and also (c) Henrik left the IETF Tools Team in late 2020, so continuity was interrupted.

Nevertheless, the opening summary from my proposal expresses the “find and fetch” use case which the IETF community lacks for its more than 8,000 specifications (referred to as “Request for Comments” (RFC). As I am the author of this proposal, I’ll reproduce a lengthy excerpt here:

> “Many specifications relevant to technical projects are maintained and promulgated by global and national standards bodies. But it is difficult for technical designers and developers to efficiently recall and to keep abreast of updates to the large number of standards that are in effect, and directly applicable to their tasks or components at any given time.
>
> We propose a collaborative project of iterative experimentation using selected standards documents of the Internet Engineering Task Force (IETF) as sample requirements (i.e. BCP, Best Current Practice documents). Factors that invoke particular requirements of a small representative sample of the standards will be expressed as computable ‘control tables’. Sample unit tests will be designed and run to simulate the normal technical work of creating, maintaining or diagnosing Internet subsystems or services. The free/libre/open “Internet of Rules” specifications and components (currently at alpha testing stage through Xalgorithms Foundation) will provide the automated rules discovery method.
>
> The purpose of the experiment will be to determine if sufficient contextual and operational data can be generated from unit tests to automatically find and return the correct technical requirements to the working developer, as needed. If this method of automated rules discovery can be made to work with some initial very simple tests, additional tests will be designed with incrementally increasing complexity and uncertainty, until a practical limit is determined.
>
> The objective of this experiment would be to determine if Xalgorithms “Internet of Rules” specifications and component may provide a practicable and effective method for designers and developers to receive a discrete, helpful interface to the specific requirement statements from the applicable standards ‘in the moment’ as they are testing some element of their implementation. This is the scenario within which they are most likely to be paying attention to conformance considerations, and will have the opportunity to act on such information to enhance conformance.” (Potvin, 2019b)

5.4 Rules as Data

5.4.1 Data Structure of [rule.dwd] Records

Earlier in this dissertation it was put forward that:

- A rule is an *imperative* statement of obligation, permission or encouragement among people;
- Documentation about a rule is a *declarative* statement as a ‘normative proposition’;
- Applicability and invocation of a rule to a circumstance is an *empirical* statement of deduction.

In the DWDS, a rule-maker agent communicates imperative assertions with normative propositions to assist rule-taker agents with empirical deductions. Obligation, permission or encouragement among human and machine ‘agents’ may be communicated with optimal efficiency in a complex dynamic multi-objective multi-constraint setting.[62] The end-to-end information transmission must be intuitive enough for a broad

62 This is unlike ‘maximum’ efficiency in terms of a single criterion.

population of human rule-maker agents and rule-taker agents to communicate normative propositions among themselves without having to know formal data processing or computer programming methods, and yet it must also structure the transmitted information precisely enough to be readily parsed and processed on any computing platform the rule-taker agents may prefer to employ or delegate to.

The default deployment of any node in a network which implements the DWDS includes all three loosely coupled components, RuleMaker, RuleReserve and RuleTaker. RuleReserve provides a passive data storage service to RuleMaker, and a passive data sifting service to RuleTaker. Users of RuleMaker applications and of RuleTaker components can have their own Subset RuleReserve nodes, or they may decide to have external third-party suppliers of RuleReserve nodes bundled with services for quality assurance, security, up-time guarantees, and error and omissions insurance. These function together as a type of data 'pipeline' (von Landesberger et al., 2017). RuleReserve receives an [is.dwd] request message from RT instances, and employs its descriptive data about a particular circumstance as a virtual sieve to sift 'in effect' and 'applicable' [rule.dwd] records from its entire collection, as well as any [lookup.dwd] tables that those rules require to operate. First RuleReserve sifts for rules 'in effect' to get an intermediate list, then it sifts again for rules 'applicable'. What remains is packaged into an [ought1.dwd] message and provided back to the requesting RuleTaker instance. At that point, RuleTaker will then sift the logic gates of each [rule.dwd] record in the [ought1.dwd] rows to determine what output assertions are actually 'invoked', and from this generate an [ought2.dwd] message that is delivered to the end-user, or their application or machine. (End users have the option to have RT run an additional round of 'in effect' and/or 'applicable' sifting operations with a revised or refined [sieve2.dwd] prior to resolving the logic gates.)

The DWDS enables three parallel representations of the same 'rules-as-data':

- General users get a graphical interface that prioritizes their comprehension of the information;
- Technical users and machines get a JSON record prioritizing data integrity and transmissibility;
- Machines get an indexed record that prioritizes storage efficiency and processing speed.

Every human-accessible [rule.dwd] record, upon commit from a RuleMaker application to a RuleReserve node, is immediately pre-parsed into directly-processable data, so that it does not have to be parsed again at compute time. A parser in RM uses a pre-defined grammatical framework to transcribe it into a hierarchical data structure, and this data is splayed out along a single row of the wide-column distributed database. In this form of storage, each row of data can be processed in aggregate with any number of other rows containing [rule.dwd] or [lookup.dwd] records. RuleReserve nodes have two functions: immutable storage and fast columnar data sifting.[63]

63 Early experimental prototyping employed MongoDB for storing JSON files, and Cassandra for fast columnar queries, which could be swapped for ScyllaDB for faster performance.

In order to accommodate input variability that can be expected from diverse rule authors on a decentralized network, it would be optimal to employ a ‘recursive descent’ style of parser configured with several parsing algorithms. In the event one method fails, it returns to the beginning of the record and attempts an alternative available parsing method. When a record cannot be parsed, a notification with diagnostic evidence is provided to the current user and to the event log. Every RR node interacts with RuleTaker (RT) clients through network data streaming that logs requests and responses.[64] The event log can be analyzed for patterns, which could indicate potential improvements.

The sifting operations of DWDS depend upon the data structure of [rule.dwd] records. We’ll use the JSON format to illustrate this, but the reader should keep in mind that a CBOR representation, and the human-optimized graphical form, and the machine-optimized horizontal tape form are concurrent and informationally equivalent. Tables 25 and 26 show the sample rule (Grocery Store Delivery Policy) in eight sections, which are coloured for clarity. The first five sections provide classes of metadata about each rule, which are used for sifting operations of the RR network. The rule logic of sections 6 through 8 are used for logic gate sifting within RT components.

Table 25: Sections of a [rule.dwd] Record in JSON

Metadata Used by RuleReserve *Data used to sift for rules ‘In Effect’ and ‘Applicable’*	**Logic Data Used by RuleTaker** *Data used to sift for rules ‘Invoked’*
1. Rule Identity	
2. RuleMaker Identities	6. WHEN these Input Conditions
3. Linked Rules or Lookups	7. THEN these Output Assertions
4. GIVEN this Context: Where and when this rule is 'in effect'.	8. Output Weights and Characteristics
5. WHEN these Categories: Activities and things to which this rule is 'applicable'.	

64 This can be implemented with Kafka, Pulsar, or equivalent.

Table 26: JSON Representation of a Sample [rule.dwd] Record


```
{
  "id": "24f44897-b6ad-4ca0-8f7d-03c059b08e86",
  "uuid": "24f44897-b6ad-4ca0-8f7d-03c059b08e86",
  "rule_id": "24f44897-b6ad-4ca0-8f7d-03c059b08e86",
  "rulereserve_nodes": "*",
  "version_standard_url": "https://semver.org/",
  "dwds_schema_version": "0.0.0",
  "properties": {
    "id": "24f44897-b6ad-4ca0-8f7d-03c059b08e86"
  },
  "metadata": {
    "rule": {
      "120_title": "Grocery Store Delivery Policy",
      "240_summary": "",
      "960_explanation": "When our standard delivery
box is more than half full and also contains at
least $100.00 in value of groceries, we provide free
delivery. This does not apply to non-standard boxes.
For all non-standard boxes, when delivery is
provided we do",
      "version": "0.4.0",
      "criticality": "",
      "url":
"https://www.grocersonline.com/deliverypolicy",
      "rulemaker_entity": [
        {
          "name": "Xalgorithms Foundation",
          "url": "https://www.xalgorithms.org",
          "uuid": "2171eb5f-a819-4ff9-bcda-
e3edb4dc7e4d"
        }
      ],
      "rulemaker_manager": [
        {
          "name": "Joseph Potvin",
          "email": "jpotvin@xalgorithms.org",
          "contact": "",
          "uuid": "0d304b5d-9c3c-4606-8abe-
45fe1835dfbe"
        }
      ],
      "rulemaker_author": [
        {
          "name": "Joseph Potvin",
          "email": "jpotvin@xalgorithms.org",
          "contact": "",
          "uuid": "78042f34-ebe1-4aa4-851c-
a0563ee1c423"
        }
      ],
      "rulemaker_maintainer": [
        {
          "name": "Joseph Potvin",
          "email": "jpotvin@xalgorithms.org",
          "contact": "",
          "uuid": "69c3e2e9-3dc1-453b-a568-
3172b80c9d18"
        }
      ],
 "linked_rules_or_lookups": [
 {
 "dwds": "",
 "column": [],
 "row": [],
 "value": []
 }
 ],
 "in_effect": [
    {
      "country": "CA",
      "subcountry": "CA-ON",
      "timezone": "UTC-05:00",
      "start": "2021-12-31T05:00:01.000Z",
      "end": "2023-12-31T04:59:59.000Z"
    }
  ],

                {
```

```
  "category_applicable": {
    "industry_classifications": [
      {
        "isic_code": "4721",
        "isic_name": "Retail sale of food in
specialized stores"
      }
    ],
    "good_service_asset": [
      {
        "unspsc_code": "78142100",
        "unspsc_name": "Logistics operation
management"
      }
    ]
  },
"input_conditions": [
    {
      "sentence":[
                {
                    "determiner":: "The"
                },
                {
                    "noun": "capacity"
                },
                {
                    "description": "of this box"
                },
                {
                    "past_participle_verb": "used"
                },
                {
                    "predicate_verb": "is"
                },
                {
                    "attribute": ">=half"
            }
            ],
            "scenarios": [
        "00",
        "01",
        "01"
            }
        },
        {
            "sentence":[
                {
                    "determiner":: "This"
                },
                {
                    "past_participle_verb":
"measured"
                },
                {
                    "noun": "box"
                },
                {
                    "attribute": "type"
                },
                {
                    "predicate_verb": "is"
                },
                {
                    "description": "standard"
            }
            ],
            "scenarios": [
        "11",
        "11",
        "01"
            }
        },
        {
            "sentence":[
                {
                    "determiner":: "The"
                },
                {
                    "noun": "value"
                },
      "sentence":[
```

```
                    "past_participle_verb":
"contained"
                },
                {
                    "description": "in this box"
                },
                {
                    "predicate_verb": "is"
                },
                {
                    "attribute": ">=$100"
                }
            ],
            "scenarios": [
        "11",
        "00",
        "01"
      ]
    }
  ],
"output_assertions": [
    {
      "sentence":[
                {
                    "determiner":: "The"
                },
                {
                    "past_participle_verb":
"advertized"
                },
                {
                    "noun": "delivery"
                },
                {
                    "description": "of groceries"
                },
                {
                    "predicate_verb": "is"
                },
                {
                    "attribute": "offered"
                }
            ],
            "scenarios": [
        "00",
        "01",
        "01"
      ]
    },
        {
                {
                    "determiner":: "The"
                },
                {
                    "past_participle_verb":
"advertized"
                },
                {
                    "noun": "price"
                },
                {
                    "description": "of the delivery
service"
                },
                {
                    "predicate_verb": "is"
                },
                {
                    "attribute": "charged"
                }
            ],
            "scenarios": [
        "00",
        "01",
        "00"
      ]
    }
  ],
  "output_weight": {
    "character": "0",
    "enforcement": "8",
    "consequences": "17",
    "rule_group": ""
  },
  "output_characteristics": {
    "ultimate_responsibility": "rule-taker",
    "primary_normative_verb": "may"
    "normative_orientation": "affirmative",
    "primary_action_verb": "to_do",
    "rule_rationale": "practical",
    "rule mood": "declarative"
  }
}
```


Any [lookup.dwd] table can be similarly represented in JSON, CBOR, human-optimized graphical form, and machine-optimized horizontal tape form without the syntax. Table 27 shows a small [lookup.dwd] table with two ISO 3166-1 country codes, and a column for the current value of the xyz_index.

Table 27: A Simple Lookup Table

		xyz_index
3166-1	CA	24.07
3166-1	CL	23.65

This can be written in JSON as follows:


```
[
 { "3166-1" : "CA", "xyz_index" : "24.07" },
 { "3166-1" : "CL", "xyz_index" : "23.65" }
]
```


When autonomous parties on a decentralized network are publishing their own [lookup.dwd] records for general use, there is a natural incentive to use standard data schemas, in effect, standard application programming interfaces (API), so that their tables will operate for the intended users.

5.4.2 Transmission Protocols for Data with Direction

The default network connection configuration of the RuleMaker, RuleTaker and RuleReserve components is “hypertext transfer protocol - secure” (https:) over transmission control protocol (TCP) port 443 for encrypted network transmission of [is.dwd] and [ought1.dwd] transitory messages, and the “InterPlanetary File System” (ipfs:) over port 4001 for network storage and retrieval of whole [rulereserve.dwd] and [lookup.dwd] persistent records to populate SupersetRuleReserve nodes. In this scenario, all messages and transmitted records mingle with general Internet traffic. An Internet of Rules can be operationalized with existing firewall and Internet traffic management settings, and network administrators have no unconventional configurations to deal with.

DWDS uses IPFS as a general-purpose resilient content delivery network (CDN), that’s to say, a geographically distributed network of servers choreographed to provide simple efficient storage and fast delivery of whole files of tabular data over the Internet for data processing on SQLite by RuleReserve nodes and by RuleTaker components.[65]

A chosen design premise of the DWDS is that data which embodies intrinsic normative direction (obligation, permission and encouragement) is a distinct class of data. One may reasonably consider whether the communication of rules might usefully shift to a network path that is dedicated to this class of data, in order to enable more effective and efficient end-user monitoring. This could be appropriate when [is.dwd] and [ought1.dwd] messages and [rule.dwd] and [lookup.dwd] resources carry data that stakeholders deem to carry significant weight for monetary, safety, security, ecological and liberty standards.

The DWDS does not require, but describes for consideration the potential for a new *‘Data With Direction Transfer Protocol’ (DWDTP)*, denoted here with the string (dwdtp:) which we suggest should transmitted over the as-yet unassigned port 7077.[66] Unconventional ports are blocked through default security configurations, firewalls and Internet traffic filters. But security-astute network administrators may find it advantageous to configure their firewalls and Internet traffic filters to accommodate transmission of this narrowly controlled class of data. The dwdtp: path is envisioned as operating natively with the QUIC messaging protocol (Quick UDP Internet Connections)[67] (Roskind, 2013) (Iyengar & Thomson, 2019) and employing Application-Layer Protocol Negotiation (ALPN) (Thomson, 2021) to automatically validate messages and data resources for conformance with DWDS requirements, as well as additional constraints the parties may want to implement.

65 The initial suggestion for our design to use IPFS came from Calvin Hutcheon, and the choice to employ it as our persistent storage method was made jointly with Don Kelly.

66 The number 7077 is currently unassigned by the Internet Corporation for Assigned Names and Numbers (ICANN). The selection of this number here is arbitrary. Should a dedicated protocol be pursued, the assigned number may differ.

67 QUIC is an new IETF standard that improves how traffic moves on the Internet. It is a transaction-oriented, minimal latency, fault-tolerant, encrypt-by-default, tunneling protocol that can multiplex a large number of request/response client-server streams with each nearly equivalent to an independent TCP connection (Transmission Control Protocol).

5.4.3 *Identifiers for [rule.dwd] and [lookup.dwd] Resources*

The DWDS requires that every [rule.dwd] and [lookup.dwd] record shall have a unique identifier so that each can be referred to precisely, conveniently and flexibly. Referring to a resource on the Internet is the purpose of the Uniform Resource Identifier (URI). In a decentralized distributed system, one cannot depend upon a central registry authority, or on coordination among parties. But perfect uniqueness is not required. It is sufficient that the probability of any particular identifier not being unique is sufficiently close to zero as to be negligible, while auxiliary resilient error management can solve for residual risk.

The practical need for a system of identifiers in the DWDS is straightforward to illustrate. A rule author composing an input condition or an output assertion within the logic gate of a [rule.dwd] record may need to draw upon data from an external [lookup.dwd] table, or may need to refer to the data output of prior [rule.dwd] record. The author might find it necessary to refer to an exact immutable set of data hosted on the server of a particular authority, or may need to call upon some dynamically updated index. In the design of a system of identifiers it is important to avoid complicating end-user or server software, and to ensure compatibility with existing schemes. (Masinter et al., 1999)

Table 28: Four Existing General-Purpose Identifier Schemes Used in DWDS Resource Identification

Identifier Names	**Uniform Resource Identifier (URI)**	**Content Identifier (CID)**	**Universal Unique Identifier (UUID) v4**	**Semantic Versioning (SemVer)**
Citations	(Berners-Lee et al., 2005)	(Benet, 2021)	(Leach et al., 2005)	(Preston-Werner, 2013)
Methods	A hierarchical sequence of named components: scheme, authority, path, query, and fragment, with naming rules	A cryptographic 256-bit string from a secure hash algorithm based on the granular data that is being identified.	A 128-bit string from a secure random character algorithm based on the contingent host's network address.	A set of three digits is assigned to represent step-wise change at three levels of significance: "Major.Minor.Patch".
Anchors	*Context Identity:* A URI is assigned to a network address. The data may be static, dynamic or versioned, and the URI does not change when the data changes. But other identity strings can be used as address components.	*Expression Identity: A* CID is derived via a consistent algorithm that uses the same data that it identifies. Even the minutest change in the data produces a different CID.	*Inception Identity:* A UUID is assigned to a static, dynamic or versioned package of data. The UUID does not change when the data or the location changes.	*Provenance Identity:* • Major changes not backwards compatible with earlier versions; • Minor changes add functionality in a backwards compatible manner; and a • Patches are adjustments that do not alter the functional design (e.g. bug fix/clarification).
Examples	With *authority component*: • ftp://**ftp.is.co.za**/rfc/rfc1808.txt • http://**www.ietf.org**/rfc/rfc2396.txt Without *authority component:* • mailto:Jane.Doe@example.com • news:comp.infosys.www.servers.unix • tel:+1-816-555-1212 • telnet://192.0.2.16:80/	Qme7ss3ARVgxv6rXtVPiikMJ8u2NLgm5szg13pYrDKEoiu	5cf24059-db02-484d-b6ac-454d6a1db707	v2.4.1

To meet these criteria, our design assembles a concise hybrid method from four existing identifiers:

- URI: Context identity (An item is recognizable by its situation.);
- CID: Expression identity (An item is recognizable by its minutiae.);
- UUID: Inception identity (An item is recognizable by its instantiation.); and
- SemVer: Provenance identity (An item is recognizable by its ancestry.)

Each of these methods is summarized in Table 28. The relationship that has been designed among them to fulfill the requirements of the identifier scheme of the DWDS is portrayed graphically in Figure 23. The resulting composite URIs may seem lengthy, but they are optimized for computational effectiveness while maintaining adequate validation through inspection, and readability of the authority and version parts.

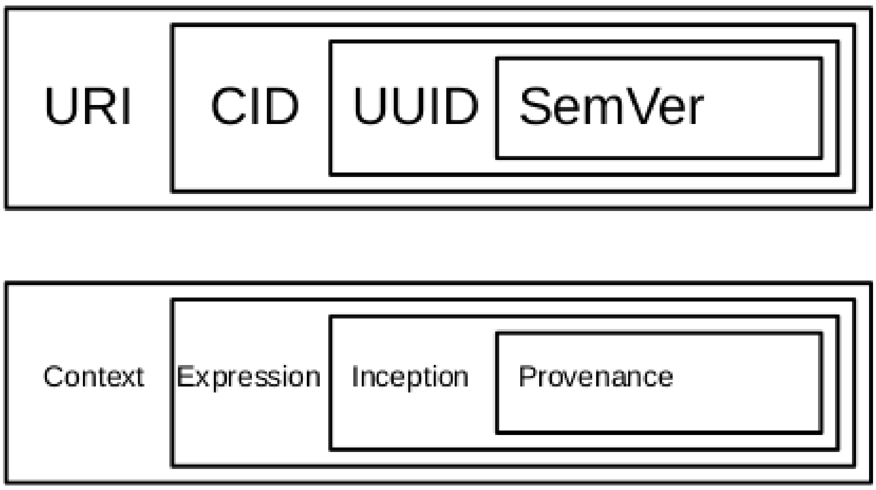


Figure 23: Relations Among the Four Identifiers as Designed for References in the Data With Direction Specification
[URI [CID [UUID [SemVer]]]]

The structure of our hybrid design for an identifier is based on the general-purpose Uniform Resource Identifier (URI) package described in the IETF standard by Paul Leach et.al. It has five parts: scheme, authority, path, query and fragment.

```
  foo://example.com:8042/over/there?name=ship#deck
  \_/ \______________/\_________________/\_________/\___/
   |         |                 |           |        |
scheme  authority            path      query   fragment
```

The URI standard does not require all of these five elements to always be used. For example the IETF document illustrates a valid Uniform Resource Name (URN) composed of just a scheme and a path:

```
urn:example:vessel:ship:deck
\_/ \_______________________/
  |              |
 scheme         path
```

The DWDS sets the ‘authority’, ‘path’ and ‘fragment’ parts as the minimum, since the fragment can embed the rest of the composite identifier. Reasons to use the other two parts might arise in later work, but current design objectives are met with just the selected three.

When a [rule.dwd] or [lookup.dwd] record is first created in a RuleMaker application, it gets auto-assigned a UUID, and the user is prompted to specify the version based on the Semantic Versioning (SemVer) structure (Preston-Werner, 2013). The UUID string is an internal part of the [rule.dwd] or [lookup.dwd] record that RuleMaker then publishes to the InterPlanetary File System (IPFS) (Benet, 2014).[68]

When the IPFS system automatically generates a ‘content identifier’ (CID) for the exact version of the [rule.dwd] or [lookup.dwd] record, this data includes the embedded UUID and the version number. The intended result is that two independent RuleMaker instances publishing the identical [rule.dwd] or [lookup.dwd] versions, will generate different CIDs due to the different embedded UUID component.

The relations of the parts used in the hybrid `dwdtp:` identifier scheme of the DWDS are shown in Figure 14. The UUID and CID identifiers are placed in the URI’s ‘path’ section, but when CID is used, this incorporates the UUID and the version number from the ‘fragment’ part. This general layout supports several ways to reference a [rule.dwd] or [lookup.dwd], in either the `https:` or `dwdtp:` schemes:

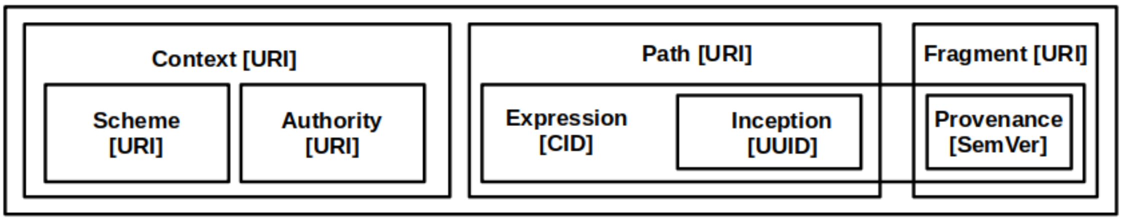


Figure 24: Components of the DWDS Resource Identification ‘Dedicated to Data with Direction’

`dwdtp:cid`	• Only the exact version of this immutable resource
`dwdtp:uuid#semver`	• A stated version of this specified resource, or the nearest available version
`dwdtp://example.org/cid`	• Only the exact version of this immutable resource from an RR node at this domain
`dwdtp:uuid`	• The latest available version of the specified resource
`dwdtp://example.org/uuid`	• The latest version of the specified resource from an RR node at this domain
`dwdtp://example.org/uuid#semver`	• A stated version of this specified resource, or the nearest available version, from an RR node at this domain

68 The Cloudflare developer website provides a useful summary of the IPFS functional design (Cloudflare, 2021): “IPFS is fundamentally a Distributed Hash Table (DHT) which maps from CIDs to people who have the content addressed by that CID. The hash table is distributed because no single node in the network holds the whole thing. Instead, each node stores a subset of the hash table, as well as information about which nodes are storing other relevant sections. When someone talks about 'uploading' content to IPFS, what they really mean (usually) is that they're announcing to the network that they have some content by adding an entry to the DHT that maps from CID to their IP address. Somebody else who wants to download their data would lookup the CID in the DHT, find the person's IP address, and download the data directly from them. The speed and reliability advantages of IPFS come from the fact that many people can upload the same data, and then downloads will be spread between all of them. If any one of them goes offline or decides to stop hosting the data, the others can pick up the slack.”

These nuanced differences support a variety of ways to reference [rule.dwd] or [lookup.dwd] records. Instead of considering `dwdtp:cid` and `dwdtp:uuid#semver` to be equivalent, RuleReserve and RuleTaker components are designed to interpret the first one as meaning *only this exact resource*; whereas the second one provides the flexibility to accept the nearest available version. And when a reference is used that leaves the version numbering unstated (null), this will default to the highest-numbered available version.

Any sentence in a [rule.dwd] logic gate can use this method to refer to another [rule.dwd] record, or to an external [lookup.dwd] table, for example

- `The posted unit price is “price”dwdtp://99b3fbb8-6e97-4b95-acfd-553a66a83741`
- `The invoice_unit_price = base_unit_price * “xyz_index” dwdtp://5cf24059-db02-484d-b6ac-454d6a1db707`

It is necessary but not sufficient that each [rule.dwd] and [lookup.dwd] be uniquely identifiable in order to be discovered and retrieved. Correct rules can be called by an identifier, but the next sections step through data sifting which is need to discover exactly what rules are ‘in effect’. ‘applicable’ and ‘invoked’.

5.4.4 Diagnostic ‘Rule 256’

Rule 256 is a system-generated diagnostic rule that is designed to be ‘in effect’ always (date/time) and everywhere (jurisdiction), for everything (good, service, asset, permission category) and every action (industry category). It has this name because it contains all 256 permutations of the full set of tetranary elements {00,01,10,11}, without nulls, in identical order for Input Conditions and Output Assertions, as shown in Table 29 and Figure 25.[69] Rule 256 is structured so that the configuration of the Input Condition scenarios matches the configuration of the Output Assertion scenarios, so that running any [sieve.dwd] on this rule should generate an output identical to the input. If it does not, then an error is signalled, supplying a simple reference to assist with real-time error-checking and after-the-fact forensic audit. RuleReserve always appends the diagnostic ‘Rule 256’ to every [ought1.dwd] response message, as a simple validation method for the integrity of the sifting operation.

69 Even technically-inclined people may overlook the tangible effect of permutations, in this case just four elements in sets of four. Table 29 illustrates this as a literal dimension. The general problem of numeric cognition of permutations is elaborated by doctoral candidate Joseph Antonides. (Antonides, 2022) Cassandra Lee of the McGill Office for Science and Society comments: “There are more ways to arrange a deck of cards than there are atoms on Earth” (Lee, 2018) The permutations of 52 cards is (52)(51)...(2)(1) = 80,658,175,170,943,878,571,660,636,856,403,766,975,289,505,440,883,277,824,000,000,000,000. (Goodman, 2014, p. 19)

Table 29: 'Rule 256' expresses in its logic gate every one of the 256 Permutations of { 00, 01, 10, 11 } in sets of four, and is structured so that the numeric configuration of the Input Condition scenarios matches the configuration of the Output Assertion scenarios to facilitate discovery of errors in the sifting process.

00,00,00,00	00,10,00,00	01,00,00,00	01,10,00,00	10,00,00,00	10,10,00,00	11,00,00,00	11,10,00,00
00,00,00,01	00,10,00,01	01,00,00,01	01,10,00,01	10,00,00,01	10,10,00,01	11,00,00,01	11,10,00,01
00,00,00,10	00,10,00,10	01,00,00,10	01,10,00,10	10,00,00,10	10,10,00,10	11,00,00,10	11,10,00,10
00,00,00,11	00,10,00,11	01,00,00,11	01,10,00,11	10,00,00,11	10,10,00,11	11,00,00,11	11,10,00,11
00,00,01,00	00,10,01,00	01,00,01,00	01,10,01,00	10,00,01,00	10,10,01,00	11,00,01,00	11,10,01,00
00,00,01,01	00,10,01,01	01,00,01,01	01,10,01,01	10,00,01,01	10,10,01,01	11,00,01,01	11,10,01,01
00,00,01,10	00,10,01,10	01,00,01,10	01,10,01,10	10,00,01,10	10,10,01,10	11,00,01,10	11,10,01,10
00,00,01,11	00,10,01,11	01,00,01,11	01,10,01,11	10,00,01,11	10,10,01,11	11,00,01,11	11,10,01,11
00,00,10,00	00,10,10,00	01,00,10,00	01,10,10,00	10,00,10,00	10,10,10,00	11,00,10,00	11,10,10,00
00,00,10,01	00,10,10,01	01,00,10,01	01,10,10,01	10,00,10,01	10,10,10,01	11,00,10,01	11,10,10,01
00,00,10,10	00,10,10,10	01,00,10,10	01,10,10,10	10,00,10,10	10,10,10,10	11,00,10,10	11,10,10,10
00,00,10,11	00,10,10,11	01,00,10,11	01,10,10,11	10,00,10,11	10,10,10,11	11,00,10,11	11,10,10,11
00,00,11,00	00,10,11,00	01,00,11,00	01,10,11,00	10,00,11,00	10,10,11,00	11,00,11,00	11,10,11,00
00,00,11,01	00,10,11,01	01,00,11,01	01,10,11,01	10,00,11,01	10,10,11,01	11,00,11,01	11,10,11,01
00,00,11,10	00,10,11,10	01,00,11,10	01,10,11,10	10,00,11,10	10,10,11,10	11,00,11,10	11,10,11,10
00,00,11,11	00,10,11,11	01,00,11,11	01,10,11,11	10,00,11,11	10,10,11,11	11,00,11,11	11,10,11,11
00,01,00,00	00,11,00,00	01,01,00,00	01,11,00,00	10,01,00,00	10,11,00,00	11,01,00,00	11,11,00,00
00,01,00,01	00,11,00,01	01,01,00,01	01,11,00,01	10,01,00,01	10,11,00,01	11,01,00,01	11,11,00,01
00,01,00,10	00,11,00,10	01,01,00,10	01,11,00,10	10,01,00,10	10,11,00,10	11,01,00,10	11,11,00,10
00,01,00,11	00,11,00,11	01,01,00,11	01,11,00,11	10,01,00,11	10,11,00,11	11,01,00,11	11,11,00,11
00,01,01,00	00,11,01,00	01,01,01,00	01,11,01,00	10,01,01,00	10,11,01,00	11,01,01,00	11,11,01,00
00,01,01,01	00,11,01,01	01,01,01,01	01,11,01,01	10,01,01,01	10,11,01,01	11,01,01,01	11,11,01,01
00,01,01,10	00,11,01,10	01,01,01,10	01,11,01,10	10,01,01,10	10,11,01,10	11,01,01,10	11,11,01,10
00,01,01,11	00,11,01,11	01,01,01,11	01,11,01,11	10,01,01,11	10,11,01,11	11,01,01,11	11,11,01,11
00,01,10,00	00,11,10,00	01,01,10,00	01,11,10,00	10,01,10,00	10,11,10,00	11,01,10,00	11,11,10,00
00,01,10,01	00,11,10,01	01,01,10,01	01,11,10,01	10,01,10,01	10,11,10,01	11,01,10,01	11,11,10,01
00,01,10,10	00,11,10,10	01,01,10,10	01,11,10,10	10,01,10,10	10,11,10,10	11,01,10,10	11,11,10,10
00,01,10,11	00,11,10,11	01,01,10,11	01,11,10,11	10,01,10,11	10,11,10,11	11,01,10,11	11,11,10,11
00,01,11,00	00,11,11,00	01,01,11,00	01,11,11,00	10,01,11,00	10,11,11,00	11,01,11,00	11,11,11,00
00,01,11,01	00,11,11,01	01,01,11,01	01,11,11,01	10,01,11,01	10,11,11,01	11,01,11,01	11,11,11,01
00,01,11,10	00,11,11,10	01,01,11,10	01,11,11,10	10,01,11,10	10,11,11,10	11,01,11,10	11,11,11,10
00,01,11,11	00,11,11,11	01,01,11,11	01,11,11,11	10,01,11,11	10,11,11,11	11,01,11,11	11,11,11,11

Figure 25: A Partial Graphical View of Logic Table in 'Rule 256' in RuleMaker's Interface

5.5 Data Sifting

5.5.1 Feature Criteria vs Conjunction Criteria

To explain precisely how DWDS data sifting works, it will be helpful to step through the process with a tangible example. Imagine a hypothetical shipping container #3009531-2261 sitting on a Halifax dock at 05:00 on 12 August 2021.[70]

Figure 26: Risk code green, yellow or red?

Before hoisting any particular container onto a ship, a dockworker needs to check and make a record of its status at that moment, and make some decisions about it. The container might be loaded or empty. It might have been inspected to validate what the manifest says. And it may or may not be locked. To complicate matters, sometimes when a container is locked, the inspection report cannot be found or retrieved, in which case the worker cannot be sure whether or nor it is empty or loaded. Or perhaps the crew on the earlier shift informally placed discarded cardboard into it for later disposal, and knowing that, the dockworker might be inclined to say that 'yes' it's loaded with recyclable material, even if it is not loaded with commercial goods. Or maybe it was loaded and inspected yesterday, but was only locked this morning, so the worker cannot be confident that the inspection remains valid. Furthermore, any particular dock worker may or may not know about some rules or recent amendments that are 'in effect' for a context, that are 'applicable' to a set of categories, and that are now 'invoked' for this particular case.

Systems modelling specialist Joshua Epstein has commented that one of the broadest purposes for creating an explicit model is to "reveal the apparently simple (complex) to be complex (simple)" (Epstein, 2008). This simple example illustrates some of the complexity which any rule system intended for general use must be designed to handle.

70 This scenario was initially prepared ahead of discussions with a data architect who works with a freight forwarding / terminal operations company at the Port de Montréal. In an online meeting in January 2021 we discussed how DWDS could potentially help with their projects involving the Digital Container Shipping Association (DCSA) *https://dcsa.org/* and the International Port Community Systems Association (IPCSA) *https://ipcsa.international/*. This example was designed to illustrate expression of logic in the context of a simple port operational concern, without complicated data relating to Bills of Lading; Packing Lists; Certificates of Origin; Bookings or Invoices.

Dock-worker Norma’s device is running a (hypothetical) operations management app *‘Opman’* with the DWDS RuleTaker component built-in, and configured to exchange specified types of data. Once authenticated, Norma enters the serial number for container #3009531-2261. Since the asset is a shipping container, Opman presents the end user a checklist of prompts.

[Y/N] This container is already tracked in the Opman application.
[Y/N] This container is placarded.
[Y/N] This container contains restricted goods.
[Y/N] This container is foreign-registered.
[Y/N] This container is loaded.
[Y/N] This container is inspected.
[Y/N] This container is locked.
[Y/N] This container is undamaged.

Figure 27: The RuleTaker assists active policy conformance.

One may suppose that the Opman app used by dock workers at the Halifax port is pre-configured to automatically supply data about the time-zone, time, date, jurisdictions (municipality, sub-country and country), and the relevant category metadata such as the International Standard Industrial Classification (ISIC) code for the “cargo transportation” industry, and the United Nations Standard Products and Services Code (UNSPSC) for “marine cargo transportation” services.

After reviewing the checklist responses and the automatically-generated data, Norma selects “Get Rules Update” on the Opman screen. This transmits an [is.dwd] package from the Opman substrate app to the RuleTaker auxiliary component. RuleTaker first runs a simple validation routine to ensure that the incoming data meets the DWDS application programming interface (API) requirements. If it does, RuleTaker proceeds to cryptographically sign and relay the data package in the form of an [is.dwd] request message to a pre-configured or random node on the RuleReserve network.

The next section below will detailed how the RuleReserve network employs the incoming [is.dwd] request message data as a [sieve1.dwd] upon the [rulereserve.dwd] collection, in order to find all [rule.dwd] records that are ‘In Effect’ and ‘Applicable’, and return them in an [ought1.dwd] response message. Then the following section will step through how the RuleTaker component uses both the [is.dwd] and the [ought1.dwd] data to tailor a [sieve2.dwd] for the logic gate of each rule, to find the Output Assertions 'Invoked', and assemble them into an [ought2.dwd] message. This resulting set of normative propositions is both human-comprehensible and machine-operational.

A useful and early comparative review of computerized rule sorting methods was published in 1987 by James Woolley and Nicholas Stone. (Woolley & Stone, 1987). For the most part, DWDS data sifting is a reformulation of sorting concepts and methods from a half century ago, from outside the field of computing. In the domain of cognitive psychology "feature search" and "conjunction search" are distinguished; DWDS employs first the former in RuleReserve nodes, and then the latter in RuleTaker components. In the 1970s, Anne Treisman and Garry Gelade differentiated feature search as being that where one's attention selects for the presence of a certain target characteristic, versus conjunction search in which one seeks a visual juxtaposition of characteristics. These authors observe: "it seems that we can detect and identify separable features in parallel across a display ... Conjunctions, on the other hand, require focal attention to be directed serially to each relevant location." [71] (Treisman & Gelade, 1980, p 132) Their observation is significant to the DWDS design. Fast parallel inspection for several individual metadata features is required in the RuleReserve nodes to determine which rules are 'in effect' and 'applicable', while a serial conjunction evaluation of data descriptors is required by the RuleTaker to determine which Output Assertion among those applicable rules is 'invoked'. The following two sections will refer to these feature search versus conjunction search methods.

5.5.2 RuleReserve Uses [is.dwd] as a [sieve1.dwd] to find [rule.dwd]s 'In Effect' and 'Applicable'

The RuleReserve network performs three functions: storage, sifting, and messaging:

- Distributed storage of [rule.dwd] records on a [rulereserve.dwd] *n* x *m* table, one record per row, maintained online via the decentralized IPFS (Benet, 2014);
- Efficient sifting to reduce [rulereserve.dwd] to a set of [rule.dwd] rows that are deemed by their authors to be 'in effect' and 'applicable';
- High-speed on-demand messaging that receives [is.dwd] requests, and sends [ought1.dwd] responses;

When the RuleReserve network receives an [is.dwd] request message, it is expected to promptly return an [ought1.dwd] response message containing every [rule.dwd] in its collection that is both 'in effect' for the context, and 'applicable' to the categories described in that originating [is.dwd] message.

71 Nancy Lobaugh et.al provide a useful summary of the two perspectives: "In a typical experiment, observers determine whether or not a specified target exists in an array of distractors. Array size is varied and the complexity of target-distractor relationships is manipulated to vary search difficulty. ... in easy feature searches, the target generally differs from the distractors on the basis of a single dimension such as colour (e.g., a search for a red bar among blue and green bars), or contains a feature not present in the distractors (e.g., a "Q" among "Os"). Search is more difficult when the target is defined by a conjunction of features present in the distractors (e.g., size and colour: a small red circle among large red circles and small blue circles) or is defined as the absence of a feature present in the distractors (e.g., an "O" among "Qs")." (Lobaugh et al., 1998)

All [rule.dwd] and [lookup.dwd] data is stored and addressed on the participating RuleReserve nodes across a distributed and deliberately redundant [*m* x *n*] matrix (i.e. *m rows* x *n columns*), referred to as [rulereserve.dwd]. The data of each indexed row is arranged like a long telex tape on which every [rule.dwd] and [lookup.dwd] record is splayed out horizontally. The sifting process may seem to be an enormous task, but it is done in massive parallel fashion across the large decentralized [rulereserve.dwd] array that is distributed on the IPFS network. The [is.dwd] message is pre-configured to function as a [sieve1.dwd] upon the [rulereserve.dwd] collection.

An example of metadata from the [is.dwd] message about the container on the dock is shown in the left-side column of Table 30, while the right-side column shows the metadata from any [rule.dwd] that would cause such rules to be caught by the [sieve.dwd].

RuleReserve node uses the [is.dwd] data to create a [sieve1.dwd], through which the metadata columns of all the rows in [rulereserve.dwd] are 'dumped', metaphorically speaking. A way to visualize this is to imagine that one needs to sort an entire truckload of blended sand, gravel and rock. One could undertake a conjunction search with an intelligent algorithm to analyze the x-axis, y-axis and z axis measurements of each particle, and have a machine place each particle into the correct pile. But a naïve feature search method involves dumping the whole truckload into an enormous mechanical sieve with vibrating screens that contain pre-determined sizes of large, then medium, then small holes. Both of these methods can sort sand, gravel and stone, but the naïve mechanical screens will get this job done faster and more economically.

The sift operation itself is a very simple, fast in-memory key-value 'feature search' on a small number of known values. RuleReserve sifts the entire holdings of the distributed [rulereserve.dwd] array in two steps.

- It first compares only the columns for date/time and jurisdiction context data. Each time these correspond in a particular row, that [rule.dwd] is deemed to be 'in effect' for the GIVEN context documented in the [sieve1.dwd], and that row's Resource Identifier is retained in memory for the second part of this sifting operation.
- The second sift is performed only in the classification columns upon rows whose Resource Identifier is held in memory. This sift operation will now only retain the rows for which available classification data of the [rule.dwd] match all of the classification data supplied in the [sieve1.dwd]. These rules are therefore deemed 'applicable' to the circumstance documented in the [is.dwd] message.

Table 30: Example: Metadata Used to Find a Rule that is Relevant to a Shipping Container at the Port of Halifax on August 2021

Metadata Used by the RuleReserve Network to Sift [rulereserve.dwd] for Rules 'In Effect' in the CONTEXT of the Port of Halifax, Nova Scotia, Canada on 12 August 2021

[is.dwd] Metadata Used in the [sieve1.dwd]	**[rule.dwd] Metadata Found by the [sieve.dwd]**

"context: [
 {
 "datetime_zone": "UTC-05:00",
 "datetime_range": {
 "datetime_attest": "2021-08-12 05:14:23",
 },
 	
 "jurisdiction_country": "CA",
 "jurisdiction_subcountry": "NS"
 }
 ],
	"context": [
	 {
	 "datetime_standard_url": "https://www.w3.org/TR/NOTE-datetime",
	 "datetime_zone": "UTC-05:00",
	 "datetime_range": {
	 "datetime_start": "1990-01-01 00:00:01",
	 "datetime_end": "2040-12-31 23:59:59"
	 },
	
	 "jurisdiction_standard_url": "https://www.iso.org/iso-3166-country-codes.html",
	 "jurisdiction_country": "CA",
	 "jurisdiction_subcountry": "NS"
	 }
	],

Metadata Used by the RuleReserve Network to Sift the Intermediate Result of the 'In Effect' Step to Obtain Rules 'Applicable' to the CATEGORY of a Shipping Container Used for Marine Cargo

[is.dwd] Metadata Used in the [sieve1.dwd]	**[rule.dwd] Metadata Found by the [sieve.dwd]**

1.
 	
"category": {
 "industry": [
 {
 "isic_industry_code": "5224",
 "isic_industry_name": "Cargo handling"
 }
 ],
 "good_service_asset": [
 {
 "unspsc_code": "78101700",
 "unspsc_name": "Marine cargo transport"
 }
	2.
	
	"category": {
	 "industry": [
	 {
	 "industry_standard_url": "https://unstats.un.org/unsd/classifications/Econ/isic",
	 "isic_industry_code": "5224",
	 "isic_industry_name": "Cargo handling"
	 }
	],
	 "good_service_asset": [
	 {
	 "good_service_asset_standard_url": "https://www.unspsc.org/",
	 "unspsc_code": "78101700",
	 "unspsc_name": "Marine cargo transport"
	 }

RuleReserve nodes actually perform this operation with the working premise that all of the records in [rulereserve.org] are considered irrelevant (recall Alfred Horn's 'proof-by-contradiction', Section 5.1), and are expected to pass right through the sieve, like sand through a coarse screen. Indeed, as soon as any one metadata element results in a 'miss', that entire [rule.dwd] row is immediately found to be irrelevant, and is eliminated from further consideration—so there is no need to compare every cell of every row. However when both the metadata and classification data fields of any rows correspond to keys in the [sieve1.dwd], this assumption of irrelevance fails. Those rows are not irrelevant, so they are captured in the sieve.

The rows that remain in the sieve after these two rounds of sifting are assembled into an [ought1.dwd] response message. This includes the full data of these rows, and each still configured in the 'horizontal tape' topology, just as they were stored in [rulereserve.dwd] on the RuleReserve network. So there are still no data transformations to perform, ensuring low error potential and high forensic auditability.

At this point, the RuleReserve node checks whether or not the [sieve1.dwd] indicates that the RuleTaker user's preference is to append chained [rule.dwd] and [lookup.dwd] tables to the [ought1.dwd] response message, and if so, what end-user constraints are specified in the [is.dwd] (reproduced in the [sieve.dwd]). This addition of chained [rule.dwd] and referenced [lookup.dwd] records is not done as the default because many users are expected to prefer to leave those for subsequent retrieval on an as-needed basis. Some [lookup.dwd] reference data files can be very large. Users could prefer to schedule the downloading of larger chained and referenced resources into their own Subset RuleReserve node(s) during off-peak hours. When indicated, the RuleReserve node checks for ResourceIdentifiers listed in the metadata of every [rule.dwd] row of the [ought.dwd] draft message. Within the end-user's conditions, these rows are fetched and appended to the [ought1.dwd] message.

Any number or rules from [rulereserve.dwd] may remain from the RuleReserve sifting operation. The results are packaged without transformation into rows of an [ought1.dwd] response message, and this is immediately dispatched back to the same RuleTaker as signed the originating [is.dwd] request to identify itself on the network as the source of the request.

The A∅ premise minimizes network operator capabilities. Logic processing to determine what rules are invoked for an end-user's circumstance is acknowledged to be a prior and exclusive prerogative of rule-taker agents. Therefore the RuleTaker components can perform this function under end-user control, which includes the option to delegate the logic processing to any third-party platform. It is a mandatory DWDS requirement that the RuleReserve network not be capable of processing the logic of the rules.

5.5.3 *RuleTaker Uses [is.dwd] and [rule.dwd], Creating a [sieve2.dwd] for Assertions 'Invoked'*

The moment a RuleTaker component receives an [ought1.dwd] response message,[72] a multi-step data process is initiated that reduces the data from rules 'in effect' and 'applicable' to achieve IS + RULE relations, to the rules 'invoked', which resolves the IS + RULE $\Longrightarrow$ OUGHT relation.

The [ought1.dwd] is comprised of a set of [rule.dwd] records arranged one-rule-per-row in a table, that are *known* to be 'in effect' and 'applicable' to the particular circumstance described in the original [is.dwd]. Now, for each of these rows, the RuleTaker component creates a transitory [sieve2.dwd] structure. Each sieves is built using subject-predicate-object triples (SPOT) data from the [is.dwd] message, and SPOT data from the sentences of the logic gate. In this section we detail how the logic gate scenarios data is 'sifted' to remove irrelevant columns, so that what remains is a set of Output Assertions that can be assembled into an [ought2.dwd] package, and offered via notification to the originating application.

There can be multiple ways to implement the DWDS RuleTaker sequence. The specification is intentionally designed to be built on diverse platforms,[73] and to remain flexible to the preferences of programmers, and to the methods available in different programming languages and mathematical notation. For our purpose here, though, comprehension can be achieved more readily with numeric and graphical representations of the nature of the sifting problem, using the example of the container on a dock.

This class of problem can become very complicated very quickly. A mere three considerations in the container example (loaded? inspected? locked?), when categorized with the four the tetranary value options {00,01,10,11}, elicits 64 potential scenarios as listed in Table 31. A selection of eight of these sets is highlighted for use in the graphical illustration on the following pages. Rarely would a rule need to express all of its possible scenario permutations of the four elements; instead, multiple sets of Input Conditions will commonly result in identical sets of Output Assertions, so multiple columns can be consolidated. But even a small selection such as these eight can be impractical to express with the methods of expression employed in other rule systems. As will be seen below, the DWDS rules-as-data structure provides an elegant and practical method to accommodate this and greater levels of complexity.

72 The previous section detailed how a RuleTaker component, upon receiving from a substrate application data that is conformant with the RuleData specification (see Section 5.3.4), immediately relays that data to the RuleReserve network in the form of a cryptographically-signed [is.dwd] request message. A RuleReserve node returns an [ought1.dwd] response message to the same RuleTaker component that previously issued the corresponding [is.dwd] request message. This [ought1.dwd] response was produced by the RuleReserve network by sifting the entire [rulereserve.dwd] compendium for only the [rule.dwd] rows which are deemed by their authors to be 'in effect' and 'applicable' to the circumstance documented in the originating [is.dwd] request message.

73 The oughtmation algorithm's data processing sequence to determine which sentences of a logic gate are 'invoked' has itself gone through several design iterations, and has benefitted from peer review of industry practitioners (Van De Ven & Johnson, 2006) (Van De Ven & Johnson, 2006), as well as partial elaboration into working reference implementations (Maranzana et al., 2008) (Maranzana et al., 2008).

Figure 28 illustrates a rule which would have a dock-worker designate a container in a port with risk code green (low), yellow (moderate) or red (significant), and to inform the active stevedore when the code is yellow or red, and, under some conditions, to co-sign a certificate for errors and omissions insurance.

Table 31: Container Loaded? Inspected? Locked? Has 64 Possible Scenarios Created by Selecting Permutations of 3 Non-Unique Items from a Set of 4 Values { 00, 01, 10, 11 }. The scenarios illustrated in Figure 28 are highlighted in yellow here.

00,00,00	**01,00,00**	10,00,00	11,00,00
00,00,01	**01,00,01**	10,00,01	11,00,01
00,00,10	01,00,10	10,00,10	11,00,10
00,00,11	01,00,11	10,00,11	11,00,11
00,01,00	**01,01,00**	10,01,00	11,01,00
00,01,01	01,01,01	**10,01,01**	11,01,01
00,01,10	01,01,10	10,01,10	11,01,10
00,01,11	01,01,11	10,01,11	11,01,11
00,10,00	01,10,00	10,10,00	11,10,00
00,10,01	01,10,01	10,10,01	11,10,01
00,10,10	01,10,10	10,10,10	11,10,10
00,10,11	01,10,11	10,10,11	11,10,11
00,11,00	01,11,00	10,11,00	11,11,00
00,11,01	01,11,01	10,11,01	**11,11,01**
00,11,10	01,11,10	10,11,10	11,11,10
00,11,11	01,11,11	10,11,11	11,11,11

Columns A through H of Figure 28 illustrate eight sample scenarios of the container example as they would appear in a logic gate of the RuleMaker interface. These columns reproduce the highlighted number sets among the 64 possibilities in Table 31.

Although humans expect full natural language sentences, the essential data unit that a declarative computer program works with is a subject-predicate-object triple (SPOT). That’s to say, when *“The container is ‘loaded’,* we have data involving a subject (*“container”*), a predicate (*“is”*), and an object (*“loaded”*). This is what makes the vertical I/O table in Figure 28 understandable, even without full sentences. Generally speaking, the DWDS algorithm relates [is.dwd] input data to [ought2.dwd] output data as portrayed in Table 32. In the RuleTaker component one SPOT (what linguists refer to as an ‘argument’) which is shown in this table with a dark background) is embedded into another SPOT (a second ‘argument’, extended with a light background). But each relation as a whole is expressed in the logic gate to to signifies a set of states, or ‘scenarios’.

		Scenarios							
		A	B	C	D	E	F	G	H
	Input Conditions								
1	container status: loaded	00	11	00	01	01	01	00	10
2	validation status: inspected	00	11	11	00	01	00	01	01
3	door status: locked	11	01	00	01	00	00	01	01
	Output Assertions								
i	risk code: green	01	01	00	00	00	00	01	01
ii	risk code: yellow	00	00	01	00	01	01	00	00
iii	risk code: red	00	00	00	01	00	00	00	00
iv	stevedore: notified	11	10	11	01	01	01	00	10
v	errors and omissions insurance: co-signed	00	01	01	10	01	00	00	01

Figure 28: A Selection of 8 Scenarios for a Container on a Dock

Table 32: Input-Output DWDS Data Endpoints

Transmitted Input Data	**Stored Input Data**	**Transmitted Output Data**
[is.dwd] message	**[rule.dwd]**	**[ought2.dwd]** message
““container status” : “loaded”” : “00”, **““validation status” : “inspected”” : “01”,** **““door status” : “locked”” : “00”,**	⟹	**““risk code” : “yellow”” : “01”,** **““stevedore” : “notified”” : “11”,** **““errors and omissions insurance” : “co-signed”” : “01”,**
Input state **“00”** means “**NO**” Input state “**01**” means **”YES”** Input state **“10”** means **”YES AND NO”** Input state **“11”** means **”YES OR NO”**		Output state **“00”** means “**NOT**” Output state “**01**” means **”MUST”** Output state **“10”** means **”MAY”** Output state **“11”** means **”SHOULD”**

Once a RuleTaker component receives back an [ought1.dwd] response message containing one or more rows of [rule.dwd] records, there are several operations to be performed before processing their logic gates.

- *RuleMaker Logs and Validates Incoming [ought1.dwd] Metadata:* The first action the RuleTaker component performs upon receiving an [ought1.dwd] response message, is to log the incoming message header data, the contained metadata and classification data. It then checks that these match the contents of the corresponding cryptographically-signed [is.dwd] message that was recently sent out. If so, RuleTaker proceeds to process the incoming data. If the [is.dwd] data does not match, RuleTaker logs the message header data from the incoming [ought1.dwd] as an 'unsolicited' message, creates an administrator notification (see "*Notify*" below), then deletes the message, and terminates the session. (Optionally, a user can configure RuleTaker to send all rejected [ought1.dwd] messages to an external storage location for analysis.)
- *RuleMaker End-User May Further Reduce an [ought1.dwd] Package*: The operator of a RuleTaker component, upon receiving a new [ought1.dwd] message from the RuleReserve network, has the option to run an additional 'in effect' and 'applicable' sifting routine in order to add or alter metadata and classification data relative to what was sent out to the RuleReserve network in the originating [is.dwd] request. There are two main reasons to enable additional metadata and classification sifting:
    - The DWDS is not premised upon rule-taker agents needing or wanting to expose all of their sift-relevant information to the distributed decentralized RuleReserve network. The rule-taker agent always retains prerogative to withhold, shroud or mask their data from exposure to the RuleReserve network. An astute rule-taker agent working under a non-disclosure constraint will generally expose just enough information to the network as would be required to obtain a wide-scope shortlist of rules. The resulting [ought1.dwd] package can then be sifted more thoroughly on the end user's own computing equipment and local secured infrastructure, using data this is restricted.
    - The moment a RuleTaker component receives a new [ought1.dwd] message from the RuleReserve network is the optimal step in the process for a rule-taker agent to screen against [rule.dwd] spam, malware and noise. The DWDS does not presume that rule-taker agents would want to have such screening imposed, since such process running in the background can be indistinguishable from censorship. Instead of re-running the 'in effect' and 'applicable' sifting routines, a user or user community could run their own Subset RuleReserve, or delegate this role to a service provider of their choice.

  This option to re-sift the [ought1.dwd] package is usually straightforward to accomplish because the default deployment configuration for every node of a DWDS network includes all three functional components: RuleMaker, RuleReserve and RuleTaker. So the received [ought1.dwd] can be staged in the user's local RuleReserve node, and modified data can be run against it to obtain a more narrowly sifted [ought1.dwd] result.

- *RuleMaker Checks Dependencies:* RuleMaker quickly scans the Input Condition and Output Assertion data of all the logic gates in the [rule.dwd] rows of the received [ought1.dwd] message to locate all Resource Identifiers of external [lookup.dwd] and [rule.dwd] dependencies. Previously the RuleReserve node will have checked the metadata of each [rule.dwd] row for these dependencies, but the DADS would not enable a RuleReserve to inspect the logic gates. Now however, RuleTaker checks the logic gate of each [rule.dwd] to determine whether any [rule.dwd] or [lookup.dwd] resources are missing from the [ought1.dwd] message, and to validate that they are available on the RuleReserve network. RuleTaker does not automatically fetch them however, unless user settings have been configured to do so; it only validates that they exist. If any of these resource dependencies cannot be found on the RuleReserve network, an alert is logged (see "Alert" below), but otherwise the rest of the process continues.
- *RuleMaker Appends Diagnostic Rule 256 (and any Additional Diagnostic Rule):* To each [ought1.dwd] response message RuleTaker appends diagnostic Rule 256. The operator of a RuleTaker component can also set the component to include additional diagnostic rules in the [ought1.dwd] package.

With all the above-mentioned preliminary steps completed, a RuleTaker component then proceeds to sift through the logic gate of each [rule.dwd] row.

Sifting a DWDS logic gate is a multi-step operation, but is quite simple. First the RuleTaker component uses data from both the [rule.dwd] and the [is.dwd] to build a transitory in-memory data structure called [sieve2.dwd]. Specifically, it reproduces in-memory the SPOTs found in the Input Condition sentences of the logic gate within the [rule.dwd] row, and then it obtains from the [is.dwd] the data required to determine the appropriate numeric states for any of the matching SPOTs it contains. *At least one* of the SPOTs of the [is.dwd] message is expected to be found among the SPOTs of the logic gate, directly or indirectly via a lookup table of synonyms. If not, RuleTaker creates an administrative notification that data is lacking in the [is.dwd], so that this rule cannot be processed. The processor then moves on to the next [rule.dwd] row in the [ought1.dwd] package.

Figure 29 illustrates the data structure required to sift through the logic gate of a [rule.dwd] record. The transitory [sieve2.dwd] data package is on the left, listing all the same Input Condition SPOTs as are contained in the logic gate. The sample rule has eight scenarios {A,B,C,D,E,F,G,H}, each permutation of the tetranary state variables: 01 (Yes); 00 (No); 10 (Yes-AND-No); 11 (Yes-OR-No). With these meanings in mind, and knowing that one should read vertically down the columns of the Input Conditions, it is not difficult to interpret the rule logic directly.[74]

The DWDS logic gate borrows the ‘hit’ and ‘miss’ terminology found in CPU cache design, where a ‘hit’ means the target data was found in the cache; and a ‘miss’ means the item was not found in the cache. Our framing of this terminology differs from the operational implication originally described by Allan Jay Smith. His definition of each ‘miss’ invoked an action:

> “When a reference (read, write, instruction fetch) is made to a cache, the reference can either find the needed information already in the cache (a hit) or a main memory operation can be required (a miss)”. (Smith, 1987, p. 2)

A transitory in-memory **[seive2.ior]** data package is built.

When a **[rule.ior]** deemed ‘in effect’ and ‘applicable is received via an **[ought1.ior]** message, the elements of its logic gate scenarios are compared with the **[seive2.ior]**.

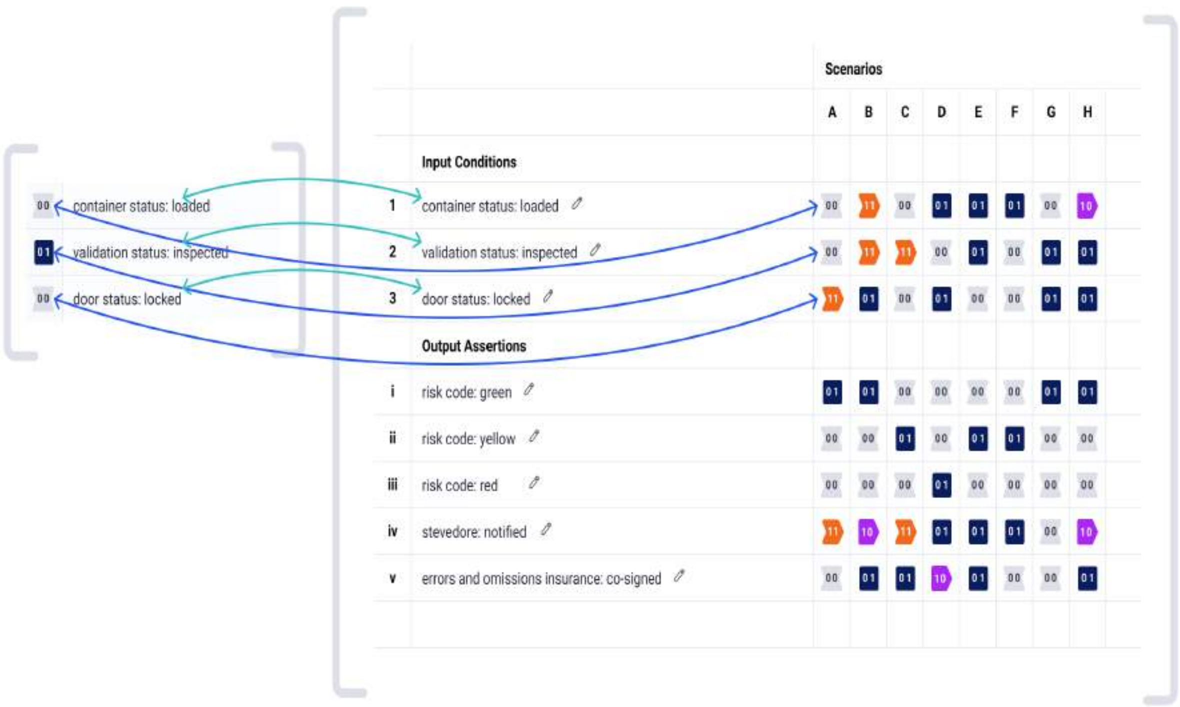


Figure 29: RuleTaker Creates an ‘In-Memory’ [sieve.dwd] from the Input Conditions of a Rule

74 The same four numeric symbols across the same three Input Conditions can support up to 64 potential scenario permutations with the four elements. But that would be a very complicated rule! A good informal assumption is Nelson Cowan’s working constraint of 4±1, (Cowan, 2001)(Cowan, 2001). For example, a well-formed rule is considered to have a maximim of three to five Output Assertions. To the extent there are more, such a rule can be workable but is expected to be less comprehensible. A rule-taker agent’s conformance with hard-to-grasp rules is probably going to be lower than for those which can be readily understood. On the other hand, when needed, the Data With Direction Specification has no prohibition against rules with many Output Assertions.

Our use of the ‘hit’ and ‘miss’ terminology for the DWDS concerns only the matching of data: where a data element from the sieve is found in the column of the logic gate there is a ‘hit’; and where a data element of the sieve is not found in the column, there is a ‘miss’. Our meaning is limited to the enabling the sifting process.

The data sifting methodology of the DWDS commences with the working hypothesis of no hits, then derives contradictions in order to deduce which ones are not false (‘proof-by-contradiction’). This adaptation of Alfred Horn’s method depends upon each of the declarative statements comprising a rule being structured in order to be falsifiable. In the DWDS, incoming observations via [is.dwd] messages can prove false that *no rules are ‘in effect’ or are ‘applicable’ or are ‘invoked’*, by locating one or more rules that are not false. This computational strategy enables fast, efficient tests for logical consequence.

The RuleTaker component now performs a simple fast “conjunction search” (Treisman & Gelade, 1980). For each SPOT it compares the numeric elements of the [rule.dwd] versus the numeric elements of the [sieve.dwd]. The RuleTaker component can be implemented do this in series, row-by-row or column-by-column, or it can be designed to perform the comparison operation in parallel upon all rows, or all columns, or indeed upon all cells of the array simultaneously. In each case, the process registers a ‘HIT’ when the numeric elements are operationally equivalent, and a ‘MISS’ when they are not.[75] For the purpose of assigning HIT and MISS tags, the elements {00, 01} are each ‘operationally equivalent’ to {10, 11}, even though they do not mean the same thing. That’s to say, a {01} maps as at ‘HIT’ when compared with a {10}, as well as with a {11}.

Figures 30 and 31 below illustrate how two particular [sieve.dwd] packages produce different sift results. Not every cell of the input conditions actually has to be compared, because the moment any cell is assigned a 00≡MISS, its entire column is known to be irrelevant to the scenario described in the [sieve.dwd]. But here we show the comparison results for all the cells of the Input Conditions.

In the first case, Figure 30, only the column for Scenario C has all of its Input Condition elements showing a ‘HIT’. This signals the RuleTaker component to proceed down that column to read the symbols provided for each of the Output Assertions. Any Output Assertions marked with 00≡NOT are ignored. The other SPOTs are invoked with 01 (MUST), 10 (MAY), or 11 (SHOULD), so they are assembled into an [ought2.dwd] message.

75 Whereas the DMN specification identifies various types of “hit policy” options (Unique, Priority, Any, First), the default being a single scenario result, our design can accommodate more than one scenario with uncertainty.

The [sieve.dwd] in Figure 30 gives the following result:

- *Scenario C: “The risk code is yellow: [MUST].”*
- *Scenario C: “The stevedore is notified: [SHOULD].”*
- *Scenario C: “Errors and omissions insurance is co-signed: [MUST].”*

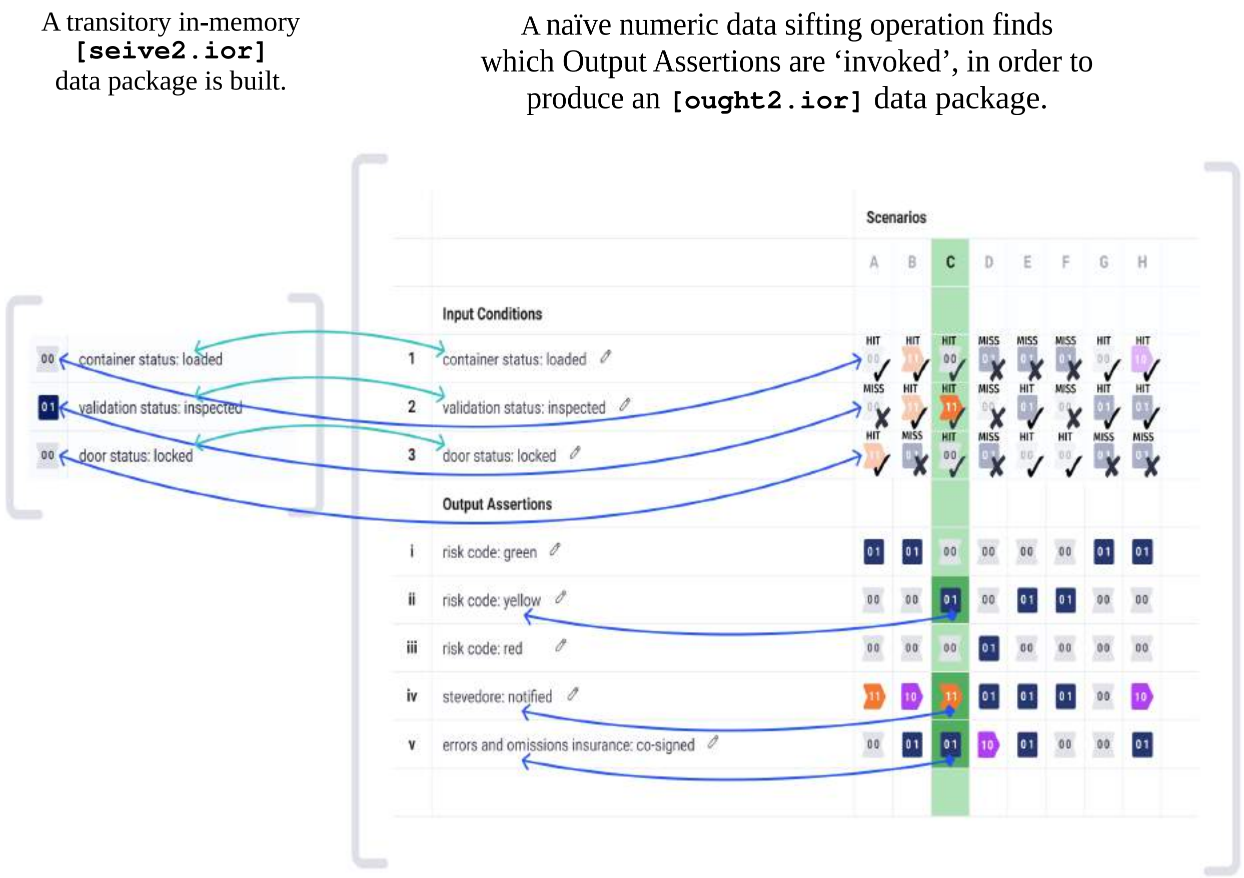


Figure 30: Example A: RuleTaker Compares the [sieve.dwd] to Elements of Each Input Condition Scenario

But the [sieve.dwd] in Figure 31 produces both Scenarios B and D with all of elements of their Input Conditions showing a ‘HIT’. When RuleTaker looks down those two columns to the symbols for the Output Assertions, and then reads the invoked SPOTs, we obtain some apparently contradictory information:

- *Scenario B: “The risk code is green: [MUST].”*
- *Scenario D: “The risk code is red: [MUST].”*
- *Scenario B: “The stevedore is notified: [MAY].”*
- *Scenario D: “The stevedore is notified: [MUST].”*
- *Scenario B: “Errors and omissions insurance is co-signed: [MUST].”*
- *Scenario D: “Errors and omissions insurance is co-signed: [MAY].”*

Figure 31: Example B: RuleTaker Compares the [sieve.dwd] to Elements of Each Input Condition Scenario

This contradictory output is a correct and meaningful result: it means there are decisions for the rule-taker agent to make. This rule is intentionally written to communicate that when a shipping container is loaded and locked, but has not been inspected, a decision-maker (human or machine) must further assess the circumstances and decide whether the risk code assigned to this container shall be green or red. If they choose risk code green, then the errors and omissions insurance must be co-signed, and the stevedore may be notified, though notification is not mandatory. And if instead they choose risk code red, in that case the stevedore must be notified, and the errors and omissions insurance is not mandatory.

Earlier in this document it was proposed that a rule is only deemed to exist when communicated. But a rule is only communicated to the extent it is recalled or found when needed, in a precise and usable form. The foregoing steps complete this circle.

In general when the resolution of a DWDS logic gate results in two or more scenarios and:

- Output Assertions of at least two scenarios are *contradictory*, then is is the role of the rule-taker agent to consider the outcome, and take one of the following actions:
    - Make a choice from among the presented options, where evidently the rule-maker intended the choice to be delegated to the rule-taker agent (reflecting 'human-centred automation'); or,
    - Report a mistake in the composition of the [rule.dwd], which could range in significance from a minor bug fix to the rule implementation, to a formal amendment to the source rule; or
    - Validate that the [is.dwd] source data that is used to build the [sieve.dwd] is correct and complete; or
    - Ignore the result.
- Output Assertions of these scenarios are *identical*, then this is due to an inefficiency in rule design, since two or more columns of its logic table could be consolidated into one column. Such an outcome is logical and usable, however it is either a sloppy rule, or this situation is indirect evidence of an error. When this occurs, the RuleTaker component requests permission of the rule-taker agent to generate an automated message to the rule-maker agents (Manager and Author) listed in [rule.dwd] metadata. As this is an efficiency not an error, there is no need for to log an alert to the RuleReserve network.

There is also the potential that the facts documented in an [is.dwd] message will result in no scenarios from the sift operation within a RuleTaker component. Such an outcome would indicate an error in the metadata of the [rule.dwd] or in the logic of the [rule.dwd] record. The prior sifting operation performed through the RuleReserve network categorized the rule in question as being '*in effect*' and '*applicable*' to the facts documented in the [is.dwd] message. Every [rule.dwd] record assembled into the [ought1.dwd] collection is expected to result in at least one Output Assertion '*invoked*'. In the event that no scenarios result from the sift operation on the logic gate, the RuleTaker component would request permission of the rule-taker agent to issue an automated message with generic information to the rule-maker agents (Manager and Author) who are listed in the metadata of that [rule.dwd] record. The RuleTaker component would also log an alert about this rule's error to the RuleReserve network. A background network service periodically scans for error and inefficiency notices in the log files, and generates automated error summaries to the Managers and Authors listed in [rule.dwd] metadata. RuleTaker components can be configured to run quality assurance [rule.dwd] records that refer to the [lookup.dwd] representation of those network log files, in order to flag [rule.dwd] records and rule-maker agents with persistent or frequent errors.

RuleTaker Performs Data Transformations if Expressed in the Rule

The [rule.dwd] and [is.dwd] artifacts consist of functionally inert declarative facts, with no procedural code. Furthermore, the DWDS requires that RuleTaker be incapable of performing any procedure that is expressed directly within an Output Assertion or that is called from a [lookup.dwd] resource. On the other hand, there is no prohibition against any such procedural function being employed by end-users, so long as they are expressed and implemented via some external, adjacent or underlying platform, never within a DWDS component or application.

The DWDS design does allow for RuleTaker to draw upon a constrained internal library of simple static cryptographically-signed methods for data transformations, Boolean operations and arithmetic functions, any of which can be employed in sentences expressing the Input Conditions and Output Assertions of a [rule.dwd] record. This would be just a small subset from the standard *ISO/IEC 80000-2:2019 Quantities and units — Part 2: Mathematics* (ISO/IEC, 2019). Such methods include multiplication and division *(a×b) ;* addition and subtraction *(a±b)*; as well as mini-max determinations *min(a, b) | max(a, b)*. These methods can be referred to with synonyms using a [lookup.dwd] (such as “>=half” in place of “>=0.5)”, but no functional methods can be added in an ad hoc way to be run within the RuleTaker component. Some common statistical functions would be included, such as standard deviation, z-score, and t-statistic (Alfieri et al., 2007). A RuleTaker component can also be configured to receive [ought1.dwd] output as flat string data only, or as multi-typed data. When multi-typed data is allowed, then for example it would parse numbers as a currency data type when they are preceded by the ASCII symbol for currency ‘¤’ and then one of the three-letter currency codes from the *ISO 4217* standard (e.g. USD; CAD) (ISO, 2015), as follows: ¤EUR2419. The formal standard currency codes can be readily extended to units defined by independent entities and distributed autonomous organizations (e.g. XRP, BTC, ETH) through a [lookup.dwd] reference.

Chapter 6: Proof-of-Concept Reference Implementations

6.1 Development of Operational Software Based on the DWDS Design

The present design research has involved incremental implementations of the four-part DWDS system specification consisting of the RuleData model, the RuleMaker application; the RuleReserve data storage and networking service; and the RuleTaker component. The first complete end-to-end specification was operationalized as Version 3.x in the first quarter of 2022, based on descriptions in the December 2021 DWD Specification (i.e. an earlier version of this dissertation) and was first demonstrated in public online sessions in March 2022. (Kelly, 2022) Refinement and interim demos in response to requests from personnel in various companies and government bodies continued throughout that year.

This chapter answers to the three design research hypotheses expressed in Section 2.3:

- H1: The problem is perceived by stakeholders.
- H2: The problem is solvable.
- H3: At least one solution to the problem can be developed and deployed with high impact.

The working implementations described in the present chapter provide a genuine demonstration that the design as of the December 2021 version of this dissertation was implementable; and evidence that the DWDS is deemed ‘useful’ (worth implementing) by some independent practitioners.

The first hypothesis, together with the broader design success criteria outlined in Section 2.4, is further addressed in Appendices C, D, E, and F with the caveat that the broader outcomes from ‘an Internet of Rules’ design can only be assessed after external deployment, in coming years.

Interpretive programming of ‘production-class’ software from a novel specification is a non-linear iterative process, requiring flexibility, attention to detail and considerable commitment among participating software developers to correct for weaknesses and fill gaps. Version 3.x of the comprehensive DWDS reference implementation has been led by two external professionals on a self-initiated volunteer basis. Addition work has been performed by an undergraduate student and a recent grad, both on paid contracts and under free/libre licenses (Apache 2.0; Affero GPL).

- RuleTaker and RuleReserve 3.x have been built by Don Kelly, an experienced full-stack developer in the private sector;
- RuleMaker 3.x was structured by Ted Kim, a senior data scientist in a public sector regulatory office.[76]

76 During part of 2021 and early 2022, Kelly and Kim were committing an average of about 15 hours per week of personal time on a *pro-bono* basis, and discussing their progress with me and other community members in open community video calls every one-to-three weeks.

- To help expedite RuleMaker application development, as of May 2022 I issued a paid full-time summer contract through the not-for-profit Xalgorithms Foundation (which I jointly set up in 2016 for such purposes) to Huda Hussain, a 3rd-year undergraduate student in the Bachelor of IT (Interactive Multimedia & Design) program at Carleton University.
- For guidance on user interface design elegance and interaction, I have also had under paid part-time contract Calvin Hutcheon, a creative technologist recently graduated from the Maryland Institute of Art (MICA).

The program source code and documentation for their implementations is posted to GitLab via the same not-for-profit Xalgorithms Foundation portfolio through which I have shared various drafts of the specification (which is this dissertation): https://gitlab.com/xalgorithms-alliance

6.2 RuleMaker at Version 3

6.2.1 RuleMaker Overview

The first full-function implementation of the RuleMaker application (Version 3.x), identified as XalgoRM inTable 11 on page 152, became available at the end of June 2022, and refinement is ongoing. This implementation of RuleMaker 3.x, started by Ted Kim, is now led by Huda. She has been relying upon four inputs:

- The latest DWDS functional specification (Chapter 5 of this dissertation in its emerging versions);
- A user interface scheme for RuleMaker led by Calvin Hutcheon;
- An implementation structure put forward by Ted Kim for Javascript frameworks;
- An earlier Version 2.0 partial implementation of RuleMaker as a client-side application created in 2020 by then undergraduate student Ryan Fleck, with contributions by student Max Chen.

RuleMaker is a limited-function IDE (Integrated Development Environment), involving a very simple form for metadata fields, the novel DWDS tetranary vertical I/O logic gate and various simple descriptive fields. Once an author enters the essential metadata about a rule, the interactive DWDS logic gate interface provides an efficient structure to compose Input Conditions and Output Assertions using using the six syntactic structures placed in any order. Semantically unconstrained declarative expressions comprising each sentence are always structured into the same six constrained syntactic elements. The author can structure all of the logic scenarios among the sentences into the adjacent columns using numeric or symbolic elements to relate the Conditions and Assertions in various permutations. A JSON viewer in the RuleMaker interface shows the author the rules-as-data expression as it is going together, which is useful both for auditing the data integrity, and for inexperience users to become familiar with the DWDS rules-as-data JSON format. The data is auto-saved during input. The user can click “Download” to save the [rule.dwd] file locally, or click “Publish” to have it automatically committed as a single row to the [rulereserve.dwd] file on a node of the distributed decentralized RuleReserve network.

6.2.2 Prior Implementations of RuleMaker

Ryan Fleck created four iterations of an experimental RuleMaker Version 1.0 Web application under a paid contract to Xalgorithms Foundation while he was a software engineering student at University of Ottawa. He joined the implementation team after attending an event that I co-hosted with Richard Stallman, President of the Free Software Foundation, about how to use the (early work-in-progress) ‘Internet of Rules’ design to solve the problem of automating payment rules in collective agreements for Canadian government employees.

The DWD Specification was incomplete and rapidly evolving when RuleMaker Versions 1.0 and 2.0 were created. Version 1.x was implemented at a stage when the DWDS design still used trinary logic for the Input Conditions {T, F, B}. However the multilingual flexibility (which depends upon the athor’s ability to change the order of the syntactic elements) was not yet designed. Nevertheless, these earlier implementations enabled our small community to test several aspects of the core DWDS data model with genuine legislative and contractual rules.

Sample rules were structured with RuleMaker Version 1.x at the request of professionals in the wider international community who learned of this design mainly through business articles by Craig Atkinson, a graduate research fellow at University of Bern, and later at Stanford Law. These references are oriented to use cases in the international trade law domain. (Atkinson, 2018a) (Atkinson & Schubert, 2021) In order to test the boundaries of the emerging method, I specifically chose sample rules that seemed complicated in their original natural language form.

In 2020 Ryan Fleck and Max Chen programmed RuleMaker Version 2.0 as a client-side application in the React/JS framework on their own volunteer time while each was working as full-time software engineers to different large private sector firms. By that time the core design described in the draft of the present dissertation had shifted to tetranary logic. I chose to represent this with the binary set {00, 01, 10, 11}. Together with Calvin Hutcheon, a set of visual icons was designed for optimal discernibility. The semantic assignments are shown in Figure 32. In the Input Conditions {00, 01, 10, 11} means {no, yes, Yes-AND-No, Yes-OR-No}; and in the Output Assertions {00, 01, 10, 11} mean {NOT, MUST, MAY, SHOULD}.

Figure 32: Semantic Assignment of Tetranary Elements for DWDS Logic Gate States

Binary	Symbol	Input Conditions	Output Assertions
00	00	NO	NOT
01	01	YES	MUST
10	10	YES AND NO (BOTH)	MAY
11	11	YES OR NO (UNSURE)	SHOULD

6.2.3 RuleMaker Version 3.x Implementation Details

In 2022 Ted Kim and Huda Hussain completely re-wrote RuleMaker as the present Version 3.x online application, based on Web forms using the Svelte/JS framework.[77] Compared with the earlier React/JS framework used for Version 2.x, the Svelte/JS environment is generally faster to write functions in, easier to learn, results in much less code, and is easier to read and maintain. All of these characteristics also enhance the intrinsic security, community potential, code longevity and cost control of an application. Svelte/JS also has several characteristics that improve execution speed.

RuleMaker Version 3.x is based on the DWD Specification after December 2021. Figure 33 shows part of the form through which metadata is entered by a rule author. The RuleMaker user interface design by Calvin Hutcheon for the DWDS logic gate has been implemented, and is shown in Figure 34. User inputs in the left-side panel simultaneously auto-generates the operational JSON code in the panels at right. Users of DWDS systems do not have to write any code, they only need to express the original natural language into the controlled natural language simple sentences of the RuleFiniteStrutureGrammar. Figures 35 , 36, and 37 illustrate the flexibility that is accommodated for rule authoring, to re-order rows, columns and the syntactic elements of each sentence.

77 Svelte was designed and produced recently by Rich Harris (Harris, 2021), a graphics editor working at the New York Times. There are several prominent Javascript (JS) frameworks to choose from, since it has been adapted to diverse requirements and styles after Brendan Eich created the original bare-bones JS in 1995. Eich wrote the original Javascript in a week and half at Netscape in order to prototype a means of making Web pages interactive (Severance, 2012). Since that time JS has been adapted with various frameworks to accommodate diverse requirements and styles.

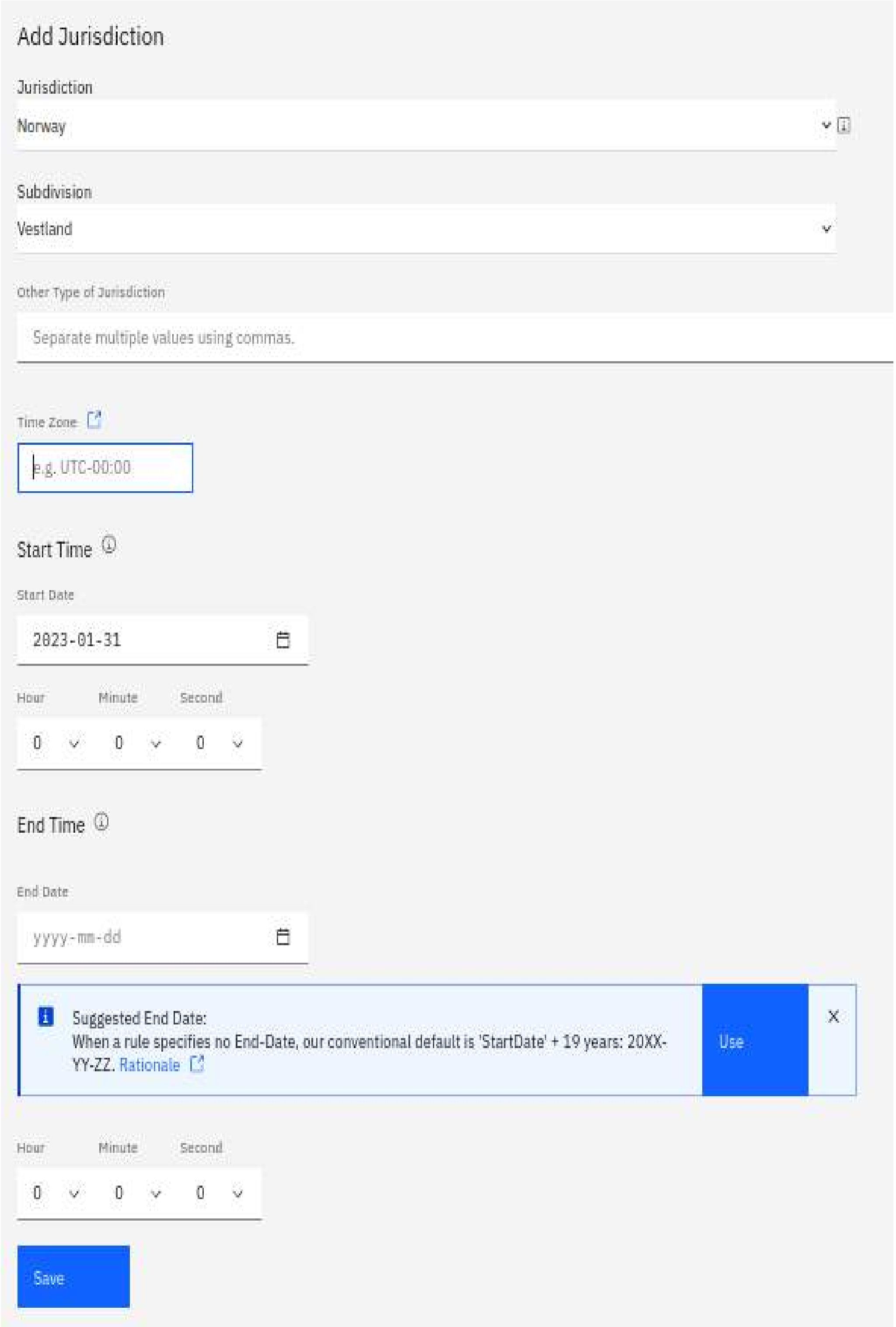


Figure 33: RuleMaker Version 3.0 prompts the author for the required metadata, and optional descriptive fields associated with each rule.

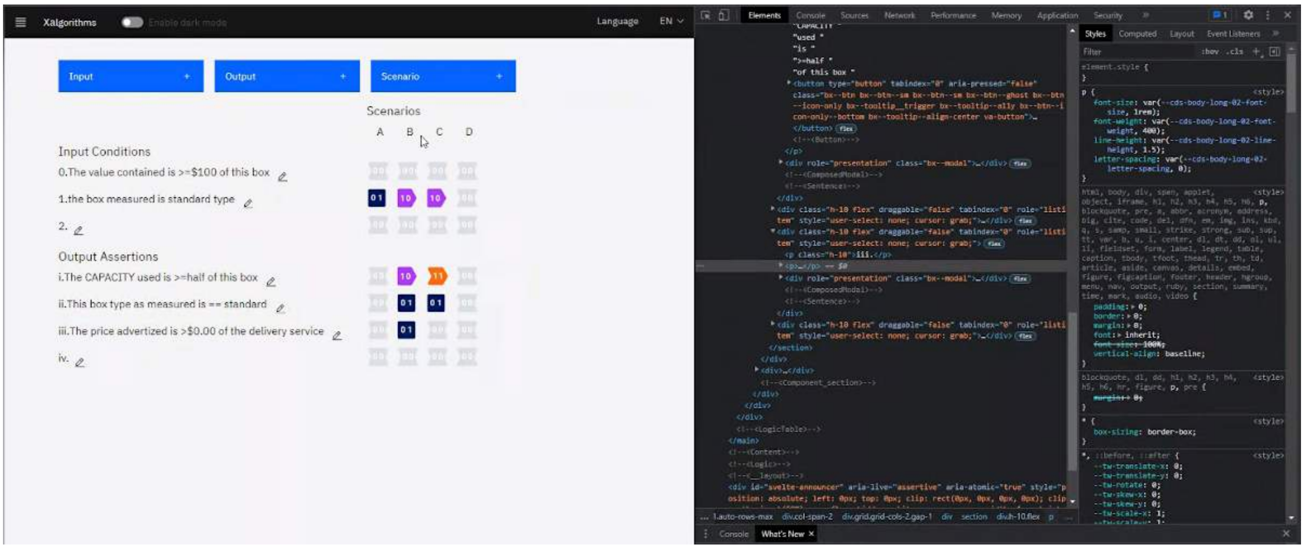


Figure 34: RuleMaker Version 3.0 automatically generates the JSON data package which is visible for the author to validate in real time.

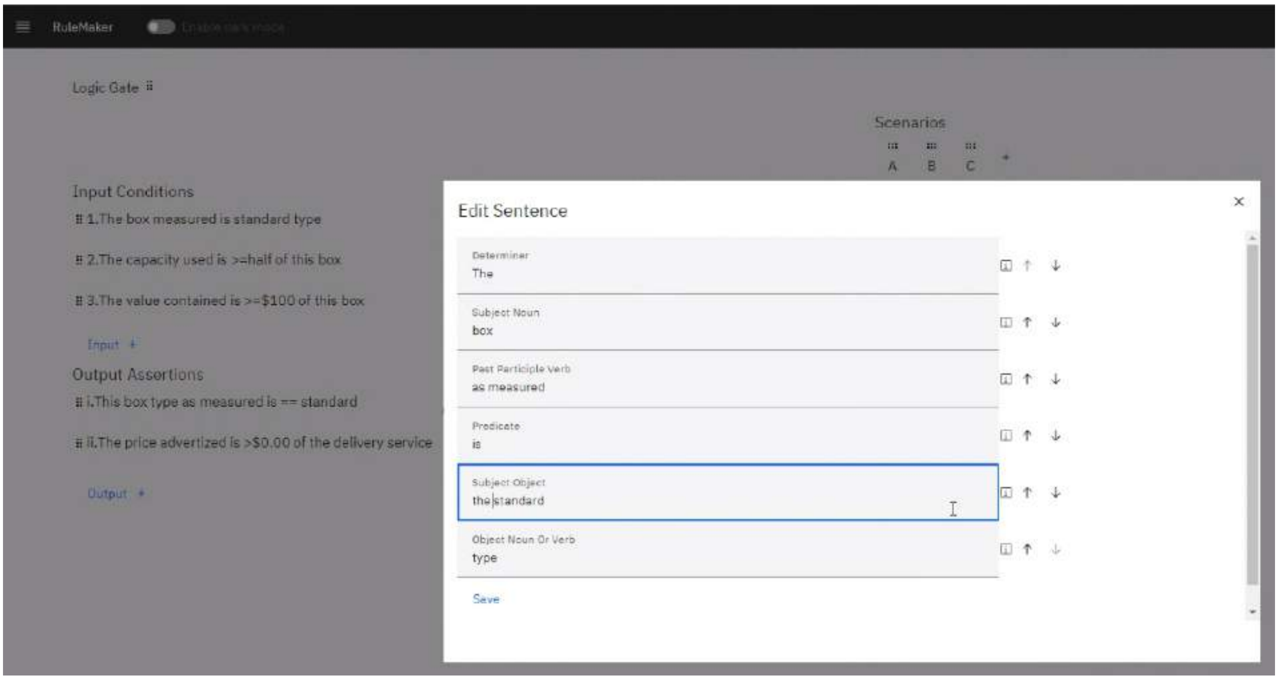


Figure 35: RuleMaker Version 3.0 provides an author the ability to change the order of the six syntactic elements of each sentence. The purpose of easy re-ordering of these elements is to help the rule author optimize the reading fluency of each sentence in any language, while also retaining the rigid syntactic constraints to enable efficient machine data processing.

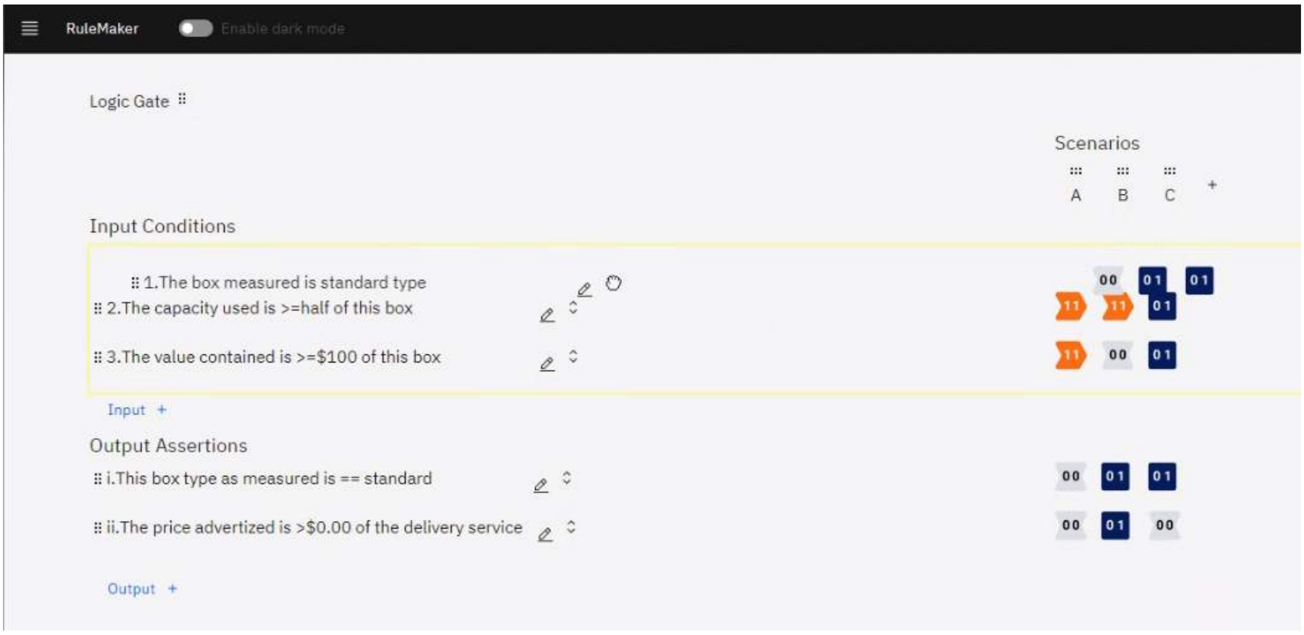


Figure 36: RuleMaker Version 3.0 lets the rule author change the order of Input Conditions and Output Assertions with a drag & drop action (illustrated here) or by up/down clicking arrows. The purpose of enabling easy re-ordering is to help the rule author arrange the logic structure of the array into an understandable configuration.

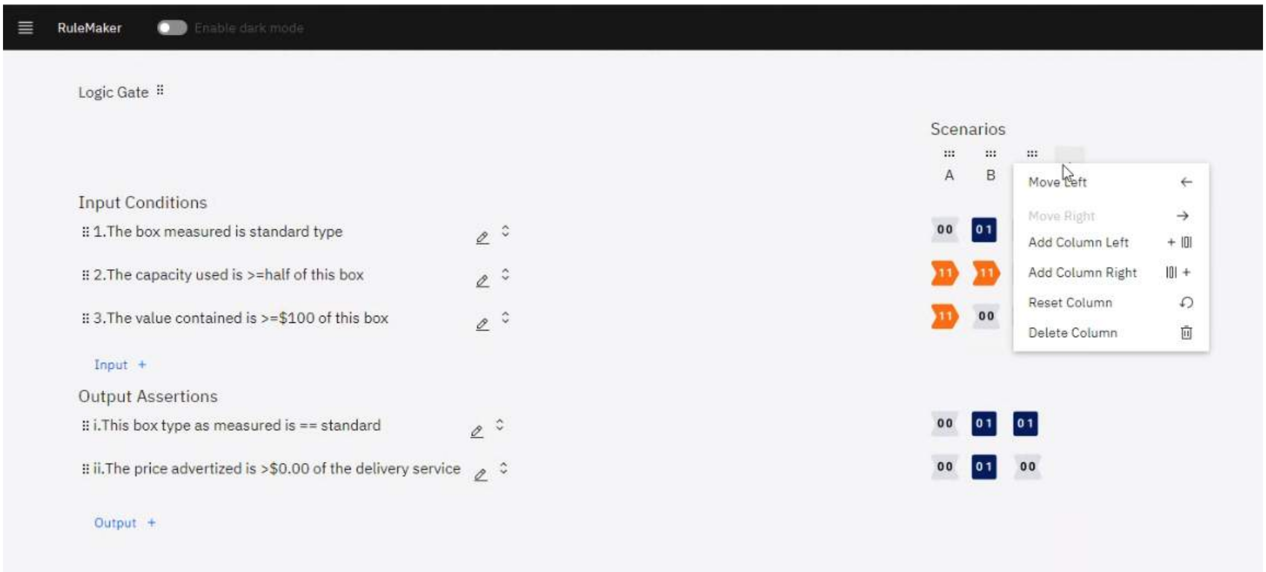
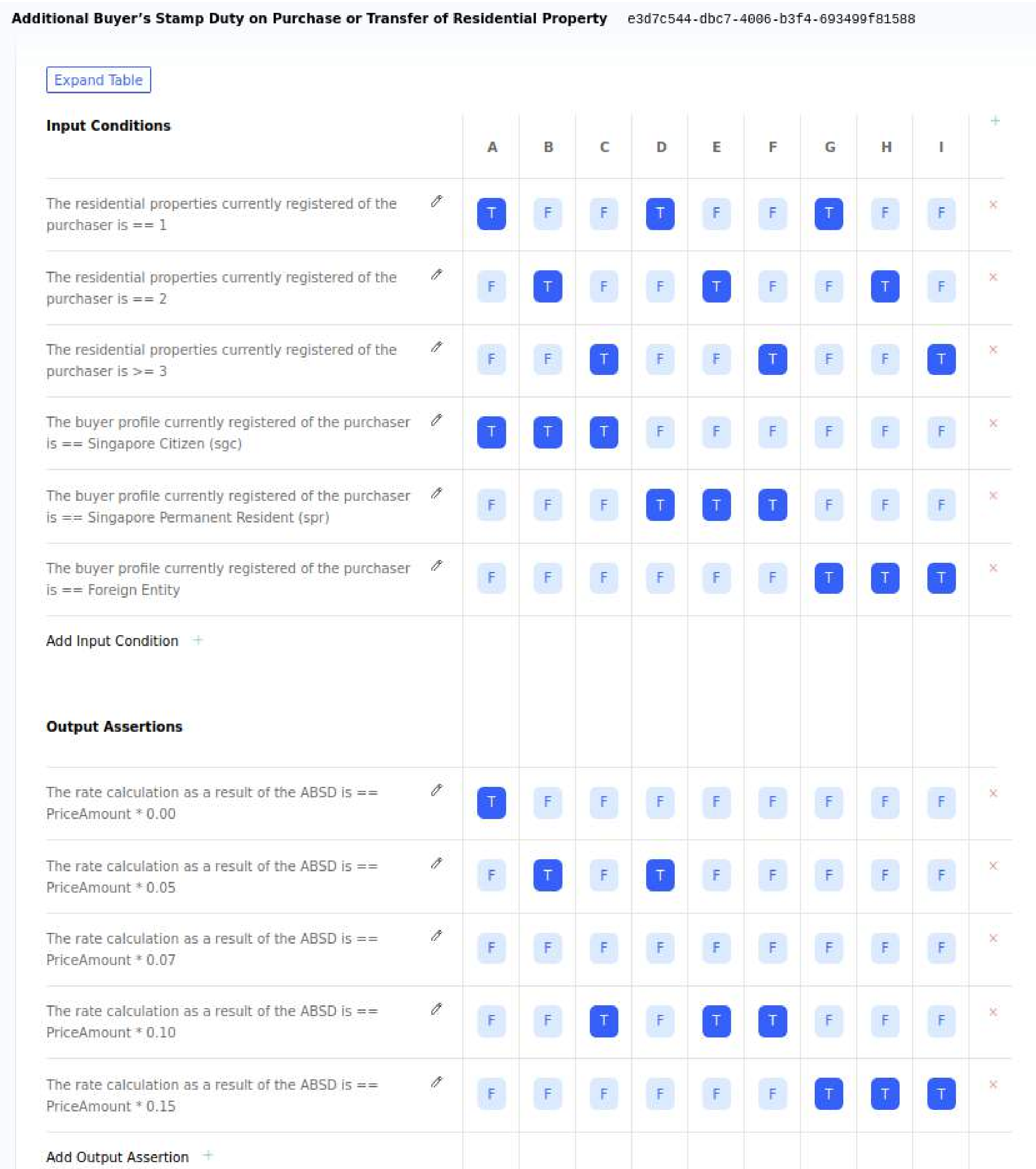


Figure 37: RuleMaker Version 3.0 lets the rule author change the order of scenario columns by left/right clicking arrows (illustrated here) or with a drag & drop action. The purpose of enabling easy re-ordering is to help the rule author arrange the logic structure of the array into an understandable configuration.

6.2.4 RuleMaker Experiments

Several real-world rules were expressed in the DWDS RuleData model through the use of RuleMaker Versions 0.x, 1.x and 2.x, while the data model was still binary {F, T} and trinary {F, T, B} where B means 'both', and while the sentence syntactic structure was not yet capable of multilingual flexibility. Following are some of the contexts for which these examples were prepared:

- In a discussions with academics and professionals at the 2018 Canada-ASEAN Business Council meeting in Singapore, March 21-22, 2018, I used RuleMaker Version 1.0 to operationalize the *Additional Buyer's Stamp Duty on Purchase or Transfer of Residential Property*, of the Inland Revenue Authority of Singapore. (See Figure 38.) The end-to-end DWDS system design was not yet completed. Now that it is, exploratory discussions have resumed with the Assistant Director, International Trade Cluster and Deputy Director (Digital/Sustainability) Ministry of Trade & Industry, Singapore.
- For a February 2019 session I organized with personnel from Canada's Treasury Board Secretariat and public sector unions, Don Kelly used a command-line precursor to the RuleMaker application, Version 0.x, to produce a sample clause that could operationalize employee pay scales from a collective agreement between the Government of Canada and the Professional Institute of Public Service Union. This involved deconstructing clauses into discrete simple sentences, but the DWDS tabular logic gates were not yet designed, so he 'black-boxed' the back-end part of the prototype to operate with a code written in a procedural style to resolve which output assertions were 'invoked' by particular permutations of input conditions.
- A November 2020 online meeting was convened by the Senior Adviser, Digital Economy & Digital Transformation, Subsecretaría de Relaciones Económicas Internacionales, Government of Chile, involving several staff and two informatics faculty members from the Pontificia Universidad Católica de Chile. I used RuleMaker Version 1.0 to illustrate the expression of a clause about national rules of origin from the Comprehensive and Progressive Agreement for Trans-Pacific Partnership (CPTPP) Article 18.50 (Protection of Undisclosed Test or Other Data) and Article 18.51 (Biologics). (See Figure 39.) Exploratory discussions continue with the Chilean government.
- At the request of the Deputy Director for Global Digital Governance, Ministry of Economy, Trade and Industry, Government of Japan, RuleMaker Version 1.0 was used to operationalize Article 12-2, Paragraphs (5) and (6) on the Regulation of Nuclear Source Material, from the Act on the Regulation of Nuclear Source Material, Nuclear Fuel Material and Reactors, Government of Japan. (See Figure 40.) Exploratory discussions continue with the Government of Japan.



Additional Buyer's Stamp Duty on Purchase or Transfer of Residential Property e3d7c544-dbc7-4006-b3f4-693499f81588

Expand Table

Input Conditions	A	B	C	D	E	F	G	H	I
The residential properties currently registered of the purchaser is == 1	T	F	F	T	F	F	T	F	F
The residential properties currently registered of the purchaser is == 2	F	T	F	F	T	F	F	T	F
The residential properties currently registered of the purchaser is >= 3	F	F	T	F	F	T	F	F	T
The buyer profile currently registered of the purchaser is == Singapore Citizen (sgc)	T	T	T	F	F	F	F	F	F
The buyer profile currently registered of the purchaser is == Singapore Permanent Resident (spr)	F	F	F	T	T	T	F	F	F
The buyer profile currently registered of the purchaser is == Foreign Entity	F	F	F	F	F	F	T	T	T
Add Input Condition +									
Output Assertions									
The rate calculation as a result of the ABSD is == PriceAmount * 0.00	T	F	F	F	F	F	F	F	F
The rate calculation as a result of the ABSD is == PriceAmount * 0.05	F	T	F	T	F	F	F	F	F
The rate calculation as a result of the ABSD is == PriceAmount * 0.07	F	F	F	F	F	F	F	F	F
The rate calculation as a result of the ABSD is == PriceAmount * 0.10	F	F	T	F	T	T	F	F	F
The rate calculation as a result of the ABSD is == PriceAmount * 0.15	F	F	F	F	F	F	T	T	T
Add Output Assertion +									

Figure 38: RuleMaker v1.0 View of Additional Buyers Stamp Duty on Property Sales, Inland Revenue Authority, Singapore

When Input Conditions	A	B	C	D	E	F	G	H	I	J		
The validated safety of the product is == required	T	T	T	T	T	T	T	T	T	T	T	
The validated efficacy of the product is == required	T	T	T	T	T	T	T	T	T	T	T	
The declared production process of the product is == from living organisms (Biologics-1)	F	T	F	F	F	F	F	F	F	F	F	
The declared components of the product is == from living organisms (Biologics-2)	F	F	T	F	F	F	F	F	F	F	F	
The declared components of the product is == a protein produced using biotechnology (Biologics-3)	F	F	F	T	F	F	F	F	F	F	F	
The declared indication OR formulation OR method of administration of the product is == novel	F	F	F	F	T	F	F	F	F	F	F	
The declared components of the product is == additional chemical entity not previously approved	F	F	F	F	F	T	F	F	F	F	F	
The declared condition of the exclusive marketing is == approval by another CPTPP regulator	F	F	F	F	F	F	T	F	F	F	F	
The active jurisdiction of the Regulator is == MYS	F	F	F	F	F	F	F	T	F	F	F	
The invoked exception of the Regulator is == Declaration on the TRIPS agreement and public health https://www.wto.org/english/thewto_e/minist_e/min01_e/mindecl_trips_e.htm	F	F	F	F	F	F	F	F	T	F	F	
The invoked exception of the Regulator is == CHL Law No. 19.039 on Industrial Property (Consolidated Law approved by Decree-Law No. 3) https://www.wipo.int/tk/en/databases/tklaws/articles/article_0005.html	F	F	F	F	F	F	F	F	F	T	F	
The invoked exception of the Regulator is == Conflict of Laws [Type 1]	F	F	F	F	F	F	F	F	F	F	T	
New Field												
Then Output Assertions												
The declared exclusive period of the product marketing is == [duration 1]	T	T	T	T	T	T	T	T	F	F	F	
The declared exclusive period of the product marketing is == [duration 1] + [duration 2]	F	T	T	T	F	F	F	F	F	F	F	
The declared exclusive period of the product marketing is == [duration 1] + [duration 3]	F	F	F	F	T	T	F	F	F	F	F	
The declared maximum application period of the EconomicOperatorParty is == within [duration 4] of approval by another CPTPP regulator	F	F	F	F	F	F	F	T	F	F	F	
The documented status of the product is == [approved] for [duration 1] exclusive period marketing by another CPTPP regulator	F	F	F	F	F	F	T	F	F	F	F	
New Field												

Figure 39: RuleMaker v1.0 View of Clauses 18.50 and 18.51 of the CPTPP Trade Agreement

Article 12-2 Paragraphs (5) & (6), Act on the Regulation of Nuclear Source Material, Nuclear Fuel Material and Reactors

Input Conditions	A	B	C	D
The licensed status of the nuclear refining activity is == "active".	T	T	T	F
The monitored days of the interval since inpsection of this nuclear refining activity is >= 90 days.	F	T	T	B
The documented material samples of the inspection of this nuclear refining activity is >= the nuclear fuel material specified in the license.	B	F	T	B
Add Input Condition +				
Output Assertions				
The regulatory inspection of the nuclear refining activity is == pending	T	T	T	F
The approved inspection of the nuclear refining activity is == inspector entry to the office, factory or place of activity.	F	T	T	F
The approved inspection of the nuclear refining activity is == inspector review of books, documents or any other necessary property.	F	T	T	F
The approved inspection of the nuclear refining activity is == question people concerned.	F	T	T	F
The approved inspection of the nuclear refining activity is == request the submission of specified nuclear fuel material.	F	T	T	F
The approved inspection of the nuclear refining activity is == request other necessary samples.	F	F	T	F
Add Output Assertion +				

Figure 40: RuleMaker v1.0 View of Article 12-2, paragraphs 5 and 6 of the Regulation of Nuclear Source Material, Japan.

- In March 2022 I was contacted by a legal researcher from the Library of National Congress of Chile, who is part of the team managing the 'World Constitutions Comparator' Web app. (Chile, 2021) The team wanted to discuss the use of the DWDS to improve human and machine access to law in Chile. In preparation for an online workshop I used RuleMaker Version 2.x to illustrate Section 92 of Canada's Constitution (with the text in Spanish). The result is shown in Figure 41.

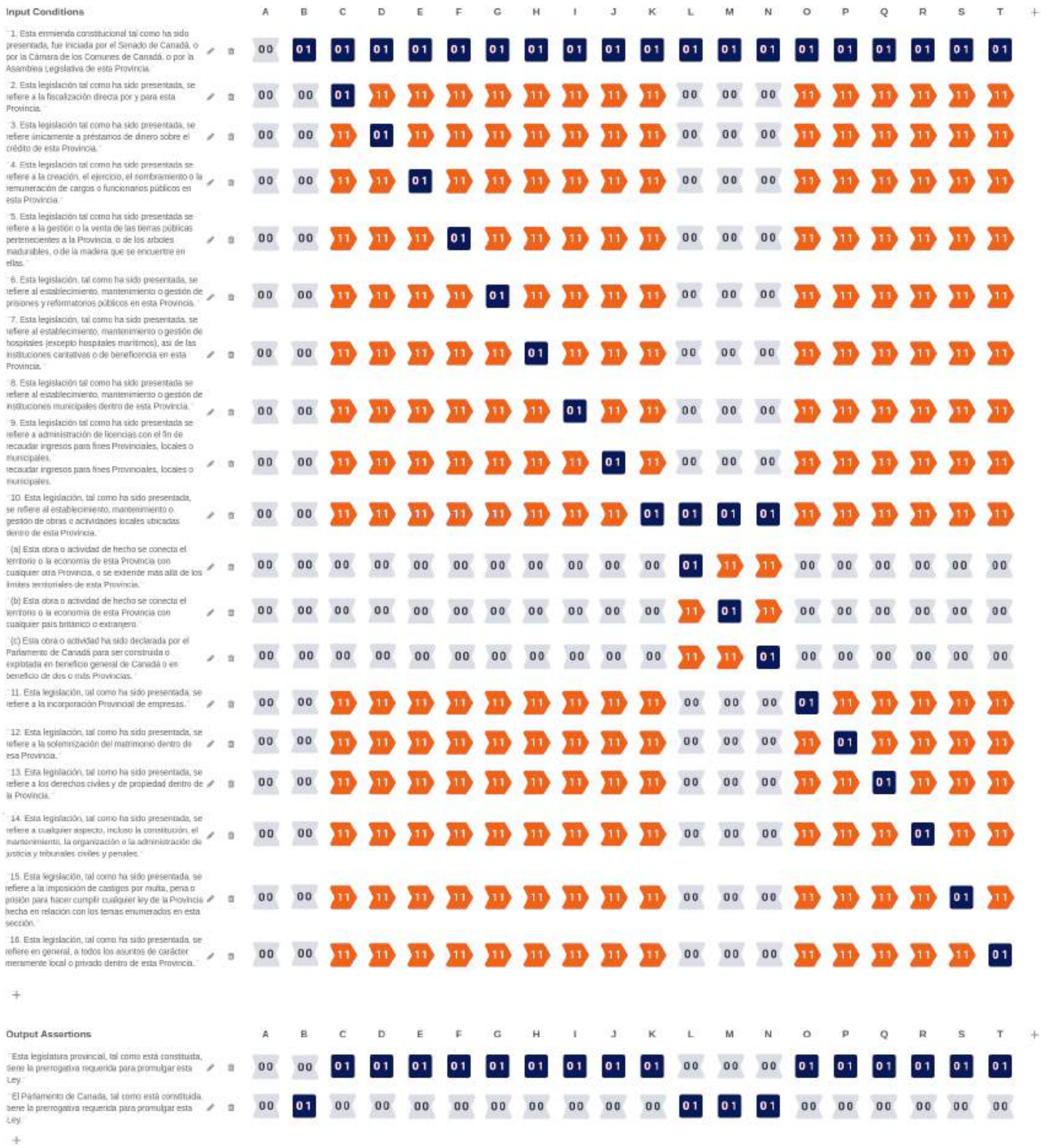


Figure 41: RuleMaker v2.0 was used to structure Section 92 of Canada's Constitution (with text in Spanish for a demo with the Chilean National Library of Congress).

6.3 RuleReserve and RuleTaker at Version 3

6.3.1 RuleReserve and RuleTaker Overview

In early 2022 Don Kelly implemented Version 3.x of the RuleReserve network service, and the RuleTaker component based on the December 2021 version of my dissertation. He built these in Ruby, and plans to rewrite them in Rust once the operational details are finalized. These are identified as XalgoRR and XalgoRT in Table 11 on page 152. Both his prototypes and the production-class implementations rely on the embedded serverless database engine SQLite.

Kelly was able to adapt, program and integrate the core capability in Ruby and SQLite on personal time over a three-month period, requiring less than 300 lines of programming code. Informally he estimates that a conventional implementation of a similar capability without the simplifications enabled by the DWDS design would have required by a team of five developers working an entire year, and would have required at least 2000 lines of programming code, with a result offering far less flexibility.

Kelly's initial reference implementation of RuleReserve Version 3.x did not require any changes to the specification, although of course he supplemented my general design with his own detailed interpretations and operational decisions, as would always be needed to produce at a working system from any specification. His source code is shared on GitLab for anyone to download, learn from, and adapt.

6.3.2 Prior Implementation of RuleReserve and RuleTaker

In 2018 and 2020 Kelly programmed parts of the system based upon the work-in-progress dissertation. Those Version 1.0 and 2.0 partial builds functioned, but at that time the DWD Specification lacked a tabular method for resolving the logic gates. On an interim basis he created a domain-specific language in the procedural style in order to resolve which of the output assertions in each rule table would be 'invoked' for a particular set of input conditions. Without a fully tabular declarative implementation style, the design did not yet enable generalizable rule authoring by non-programmers with only controlled natural language. Users would need to learn specialized expressions, and machine platforms would require an installed parsing program. I thought the programmers implementing the design would have an existing, obvious method to resolve the logic gates I had structured with a vertical I/O table, but they did not.

Finally in July 2021, literally by laying out the data structure with sticks, stones, shells and seed pods while taking a break by canoe to a tiny island in the middle of the Gatineau River, I managed to design a simple data 'sifting' method which is now described in Chapter 6 of this dissertation. This method is a restructuring and extension of the 'data kinetics' technique that I had been learning since 2016 from William Olders and Wayne Cunneyworth (Cunneyworth, 1994). My revision of their system provided a strictly tabular declarative way to determine the 'invoked' Output Assertions. It requires that RuleReserve has previously performed the simpler process of sifting the metadata to discover which rules are 'in effect' and 'applicable'. This opened the way for implementation of the end-to-end data processing pipeline in a fully tabular-declarative style to satisfy the DWDS design requirements for an Internet of Rules.

6.3.3 RuleReserve and RuleTaker Version 3.x Implementation Details

Don Kelly's operational implementation of RuleTaker and RuleReserve was demonstrated in two public online sessions in March 2022. (Xalgorithms Foundation, 2019-2022) His illustrative use case involved a common type of event-driven application for e-commerce interactions among merchants, customers, payment services and delivery services. This type of application functions by means of an 'event handler' that monitors for signals of activity, logs simple descriptive notations about them, and responds to particular event notations by executing subroutines previously specified by the programmer.

This implementation associates various 'DWD events' with the RuleTaker subroutine in the following sequence which was elaborated in this way by Don Kelly based on the December 2021 version of the present dissertation (Xalgorithms Foundation, 2021-2022):

1. A user performs an action through any underlying application interface, causing 'an event';
2. Data about that event is recorded in the event log;
3. The event handler process of the underlying application:
   (a) Detects the event signal and its context;
   (b) Matches the event signal to one of the programmer-specified 'DWD event' signals;
   (c) Matches the DWD event signal to the programmer-specified RuleTaker subroutine;
   (d) Executes the RuleTaker subroutine;
   (e) The RuleTaker subroutine:
      - Accepts active data offered to it, given the application programmer's decisions;
      - Creates an 'is.dwd' data package wrapped in standard messaging metadata;
      - Sends the [is.dwd] message to RuleReserve via localhost: or https: port 443;
      - Waits for RuleReserve to sift for 'in effect' (context) and 'applicable' (categories) rules;
      - Receives an [ought1.dwd] message (0..* rows) of containing 'rules-as-data';
      - Sifts [ought1.dwd] logic gates with [is.dwd] data, to get residual of 'invoked' assertions;
      - Creates in-memory [ought2.dwd] data package wrapped in unique reference metadata;
      - Signals the application that the [ought2.dwd] package data is available for use.

It is useful to consider this design relative to other types of request-response systems in informatics. Some common ones are:

- An ‘Interrupt I/O’ function within a device cues an ‘Interrupt Service Routine (ISR)’ to fetch known data from elsewhere in the same (or another) device (Fan, 2015, pp. 86-117).
- An SQL ‘SELECT’ command in an application can be used to formulate a local or remote database query to fetch known data from elsewhere in the same (or another) device
- Dynamic peer-to-peer systems perform lookups across sub-networks and super-networks (Stoica et al., 2003) (Korzun & Gurtov, 2013).
- Algorithmic (‘smart’) contracts use the Chainlink distributed ‘oracle’ network to fetch external data from validated sources (Ellis et al., 2017) (Breidenbach et al., 2021).

The DWDS differs from all of these in several ways:

(a) Whereas these systems fetch *positive facts* (factual declarative ‘is’) data, DWDS is tailored to fetching *normative propositions* (advisory declarative ‘ought’) data;

(b) DWDS is workable and useful whether or not anyone knows ahead of time what data results to expect for a particular circumstance. Other rules engines require that the system designer or the end user knows which rules are ‘in effect’, ‘applicable’ and ‘invoked’ for each circumstance;

(c) The [ought2.dwd] response may persuade a recipient of the information to modify the computational operation This provides a platform-agnostic means to operationalize acquired passive logic;

(d) DWDS does not require creating relational database tables for data that each rule depends on, as table lookups can be directed to records on the local or destributed RuleReserve nodes;

(e) DWDS avoids ‘hard-coding’ application pages or forms for particular rules or rule versions, therefore updates to rules do not require adjusting application pages or forms;

(f) DWDS offers consistent rule expression across platforms (data model, algorithms, libraries);

Kelly’s deployment embodies the DWDS requirement that any implementation must adopt (and must not degrade) the data security model of the substrate application architecture. His implementation demonstrates how the RuleTaker component cannot force the parties (e.g. merchant; platform operator; payments operator) to reveal to other layers the data within their respective layers. Each stakeholder can choose to share any information to the extent they prefer, and programmers of the underlying application can design their system to move data among application layers. But in Kelly's reference implementation, the RuleTaker subroutine only has access to active data within the layer where the DWD event occurs: e.g. the merchant's online storefront; or the underlying commerce platform; or the payment service. This is not a mere guideline, it has been an intrinsic requirement in the DWDS specification from the beginning.[78] However it became operationally explicit and verifiable during Don Kelly's implementation. The following points are adapted from Kelly's comments during a public demo:

- A RuleTaker subroutine MUST be incapable of actively pulling data from an application it is attached to. It is limited to passively receiving data for an [is.dwd] message that it may send to the RuleReserve network.
- A RuleTaker subroutine MUST be incapable of actively pushing any data to the application it is attached to. It is limited to offering [ought2.dwd] data to the application layer.
- A RuleTaker subroutine MUST NOT force, enable or obstruct any application layer with regard to sending, revealing or observing active data any other application layer.
- The designer(s) of each application layer MUST be empowered to determine what any application layer is capable of with regard to sending, revealing or observing active data relative to any other application layer, taking account of privacy, confidentiality and security rules of the jurisdictions of the end-users providing the data.
- The RuleTaker subroutine MUST be restricted to one method of sending data to an external destination: it sends DWDS specification-conformant [is.dwd] messages to RuleReserve nodes.
- The RuleTaker subroutine MUST be restricted to one method for obtaining data from an external source: it may receive DWDS specification-conformant [ought1.dwd] messages from RuleReserve nodes. Any other RuleTaker data transmission 'event' is flagged as an error.

To illustrate conformance of the RuleTaker reference implementation to the substrate application's data security model, Kelly proceeded from a merchant's context (shopping cart) to an e-commerce platform's context (choice of payment method). In that separate layer, when the event handler runs the RuleTaker subroutine it only has access to the active data accessible within the platform operator. This includes the merchant's identity and the customer's identity, so the platform operator may put in place some conditions for which it can select for display several payment options. But the payment layer has no need for no any access to information on what is being purchased: literally, it is not *their* business.

78 Since 2016 drafts sections of this dissertation I have described how RuleTaker components would be deployed in relation to applications using the analogy of a lichen. In nature, lichens attach to the surfaces of trees and rocks, using these only as a substrate without interfering with their internal integrity. This concept complemented the design intent that it be passive.

Over a period of five years while the DWDS design was emerging, Don Kelly and I stepped through multiple iterations of the general architecture and implementation possibilities for an Internet of Rules data processing pipeline. Initially around 2018 Kelly implemented a working version of the my early design of the RuleReserve using MongoDB to store immutable records, while using Cassandra to stage and compute dynamic data, using the map/reduce design pattern. (Dean & Ghemawat, 2008) However we both felt that this client/server approach imposed unnecessarily heavy set-up and maintenance overhead, so we made repeated efforts to reduce the technology requirements to the absolute minimum essential to achieve the intended functions.

In November 2020, in consultation with Kelly, I came up with the idea of SupersetRR nodes and SubsetRR nodes while modeling a set of sequence diagrams to aid my thinking through the end-to-end system. The SubsetRR nodes greatly simply the run-time process because distributed pre-selection has already occurred.

Then, through summer of 2021, in consultation with Wayne Cunneryworth and William Olders I re-designed nearly from scratch the data selection method. Kelly and I resolved to call this process “data sifting”, and we determined that this sift process should first occur within each RuleReserve node using only the rule metadata to sift for ‘in effect’ and ‘applicable’. A separate second round of sifting would occur within RuleTaker nodes, using only the logic gates to identify which parts of ‘in effect’ and ‘applicable’ rules are ‘invoked’.

The re-design was a great success. Wayne Cunneyworth and William Olders responded by saying this resulted in a more efficient method than their own design that is currently used in centralized deployment on mainframe systems for large financial companies—which I was only attempting to mimic in a decentralized way.

Kelly found that he could now meet all the data staging and processing requirements of RuleReserve and RuleTaker using the SQLite ‘localhost’ embedded serverless database engine. To help explain the utility of SQLite for our deployment context, he pointed me to an interview by Mahdi Yusuf with Simon Willison:

> “SQLite is often underestimated by developers. It provides a modern, extremely fast and extremely well tested relational database engine, and it comes bundled as part of Python in the sqlite3 standard library module. It effortlessly handles many GBs of data and includes powerful features like JSON support and full-text search. Importantly, a SQLite database is contained in a single binary file. This makes them easy to copy, share and upload to hosting providers. The cognitive overhead involved in working with SQLite databases is tiny: create or download a .db file and you're ready to go. ... My breakthrough was the realization that read-only data packaged as a SQLite database could be deployed to inexpensive serverless hosting, where it could scale from zero (costing nothing) up to handling an unlimited amount of inbound traffic.” (Yusuf, 2022)

The decentralized and distributed Inter-Planetary File System (IPFS) could then be relied upon to store the immutable [rulereserve.dwd] versioned records. IPFS is a general-purpose ‘content delivery network’ (CDN), which is to say, a geographically distributed network of servers choreographed to provide fast delivery of Internet content. It delivers whole files, and is not for queries. The initial suggestion for our design to use IPFS came from Calvin Hutcheon, and the choice to employ it as our persistent storage method was made jointly with Don Kelly.

With these implementation choices we found that our set-up and maintenance overhead had been reduced to the point that the default deployment for every DWDS node could include all three elements: RuleMaker, RuleReserve and RuleTaker. This provides an attractive model of decentralized node neutrality for a genuine “Internet of Rules”. Anyone operating a node could alternatively choose to just implement one or two of the core components of DWDS to meet the particular requirements. Typically an IoT device would need only RuleTaker, without a RuleMaker client, and multiple IoT devices can share an external dedicated SubsetRR.

Kelly’s current reference implementation of RuleReserve 3.0 in Ruby with an SQLite ‘localhost’ is operating as an auxiliary service adjacent to an e-commerce application substrate. Presently he is adding the ability for SupersetRR nodes to pull [rulereserve.dwd] files from IPFS, and then will enable these nodes to exchange updates on a peer-to-peer basis. Details regarding how these version updates will occur await his implementation effort to learn exactly how IPFS notifications work. But essentially, each SupersetRR node would monitor for updated versions of the [rulereserve.dwd] that it ‘subscribes’ to on IPFS – each immutable version of a [rulereserve.dwd] file has a different CID (content identifier). When a SupersetRR downloads the latest [rulereserve.dwd] file, it would compare it with the previous version it already has, create a ‘diff’ table, and automatically broadcast the differences to all of its subscribed SubsetRRs. The various SupersetRRs peers would also validate the diff tables among each other, checking that the differences they each detect between any two versions map exactly. This is a useful routine integrity check. The default scheduled frequency of these [rulereserve.dwd] updates can be just once-per day. However any number of independent specialized methods can separately perform urgent updates to any SubsetRR, on any commercial or not-for-profit basis as market participants choose to support.

Anyone may write their own independent RuleReserve implementation based on the DWDS specification using other components. For example the operator of a Superset RuleReserve may want to use a Cassandra database instead of SQLite. In our assessment SQLite is adequate for the job because the sift process I designed is exceedingly simple. No partitioning or sharding is required because the sift procedure is limited to comparing atomic data. The sift method requires no JOIN statements, there are only some rudimentary SELECT and WHERE statements. SQLite indexes the [rulereserve.dwd] table on the columns used to select for 'in effect' and 'applicable' rows. The most complicated comparison the RR sift procedure performs is dates.

6.3.4 RuleReserve and RuleTaker Experiments

Several rules were made operational in Don Kelly's e-commerce reference implementation of RuleTaker and RuleReserve Version 3.x in order to have an algorithmic shopping cart dynamically find and fetch rules(e.g. rebates, product notices, price discounts, and delivery criteria). It will be sufficient here to consider in detail just one of his tested rules in order to highlight the essential operational characteristics of the implemented design. This sample involves a genuine regulatory requirement with an exemption. The rule is maintained on a RuleReserve node, but it is dynamically discovered, so that then the retail shopping cart process is able to use it.

For this test, Kelly used some elements of Sections 40(1)a and (2)i and Section 141(3)1 of the "Licensing, Regulation 746/21" under Ontario's *Liquor License and Control Act* (Ontario, 2021). These regulations came into effect in 2021, and are applicable whenever beer or liquor are sold retail in the province of Ontario:

> 40. (1) Despite subsection 39 (1) (Removal of liquor from licensed premises), and subject to subsection (2), liquor in a securely closed container may be removed by a patron from the licensed premises if the licensee ensures that,
>
> (a) the liquor is purchased together with food sold by the licensee at the licensed premises;
>
> (b) the food and liquor are removed from the licensed premises together; ...
>
> (2) Subsection (1) does not apply with respect to a licensee in the following circumstances:
>
> i. the primary purpose of the premises is a purpose other than the sale and service of food or liquor for consumption on the premises"
>
> and
>
> 141. (3) The holder of a brewery license may use a third-party service provider to distribute its beer only if one of the following conditions is met:
>
> 1. The third-party service provider does not distribute any products other than liquor manufactured by the holder of the brewery license or its affiliates as part of the shipment in which the beer is distributed.

In a ‘localhost’ test instance of RuleReserve, Kelly expressed in JSON, conformant with DWDS RuleData, the default part of section 40(1)a: “purchased together with food”, as well as the exception, section 141(3)1: “third-party service provider does not distribute any products other than liquor manufactured by the holder of the brewery license”. Together these two parts of the regulation have the logical result that while craft breweries can sell their own beer to walk-in customers by the single unit or by the case, any beer they sell which is made by *other* suppliers can only be sold retail together with food.

- During the demo of the e-commerce reference implementation, the craft brewery’s own products could be added to the shopping cart without any rule showing up. Behind the scenes within RuleTaker, section 141(3)1 was overriding section 40(1)a. But the moment a third-party beer was added to the shopping cart, RuleTaker supplied section 40(1)a to the substrate application. This is because section 141(3)1 was no longer over-riding 40(1)a. In this simulation the developer of the substrate application had previously tested for rules ‘in effect’ and ‘applicable’, and arranged for the reloaded page to display to the user a selection of chips to choose from within the order page. None of the logic has to be programmed into the shopping cart, rather it all gets dynamically generated in a split-second response. Only *the merchant’s response to the outcome of the logic* is programmed in the merchant’s layer, which is to display a sub-menu of whatever food products they currently have.

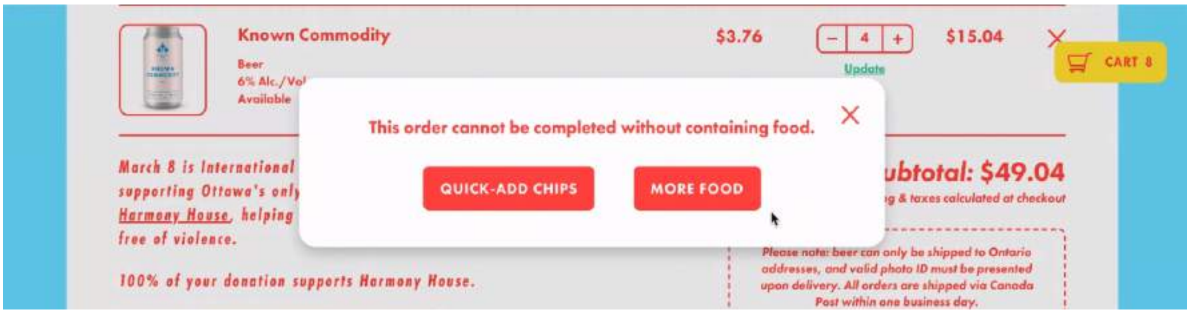


Figure 42: RuleTaker tested with a government regulation concerning third-party sales.
In this test, the platform programmer implemented RuleTaker to run as an invisible auxiliary service to the host e-commerce application. This enabled a hypothetical merchant to easily configure their shopping cart to provide an [is.dwd] data package to RuleTaker each time product is added. RuleTaker immediately fetches any 'in effect' and 'applicable' rules from RuleReserve network, receiving back an [ought1.dwd] message, which it then locally sifts to find the rules 'invoked'. Though the merchant may not have known previously about this provincial regulation, testing would have shown them to anticipate a regulatory requirement to bundle some of their products with food. Therefore in this example the merchant knew to set up an integrated dialog box within the host application, so that whenever this class of rule showed up, the shopping cart will display the merchant’s current food selections to the customer. Should the province ever rescind the rule, the merchant would not have any work to do, since the dialog would simply not not be triggered.

- A merchant who did not previously know about this particular rule could have simply had the resulting text of the rule displayed as information to the customer. However once the merchant does know of such a rule, they can configure the shopping cart application so that whenever a product “must be sold with food”, a dialog bog displays an auxiliary menu of food products that the customer can select from, as Kelly demonstrated. The event handler in this demo was configured by the developer of the substrate application so that during every page load, all the active data within the layer would be packaged into the ‘is.dwd’ message, in order to find out what rules *might* be ‘in effect’ and ‘applicable’. *No particular rules were assumed*.

- The DWDS design and its implementation methods have to facilitate extremely fast processing. The [ought2.dwd] package must be returned almost instantly after the [is.dwd] message is sent, so that RuleTaker can support dynamic responsive behaviour, even while a users is entering data into a field. In this demo, each time a product is added to the shopping cart, in a fraction of a second the RuleTaker’s subroutine executes so that it:
    - Receives available active data presented to it;
    - Sends an ‘is.dwd’ message to the RuleReserve;
    - Receives back an ‘ought1.dwd’ message;
    - Sifts the received logic gates; and,
    - Presents the residual ‘ought2.dwd’ to the substrate application.

  This request-response speed supports interactive behaviour which enables the merchant to integrate finding and fetching of external rules dynamically into their ongoing engagement with the customer.

- A few observations can also be offered with regard to some other rules that were shown during the demo in March 2022. Kelly proceeded through the shopping cart stage of the application to the check-out. When he typed the buyer’s location into an optional field, immediately some new rules arrived. One of these was the merchant’s own rule that qualified this order as eligible for free delivery at the entered address. As in the previous example this logic was not programmed into the checkout, but was dynamically generated from an interaction via RuleTaker to the localhost: RuleReserve. A merchant selling a certain product locally, and selling the same product across jurisdictional boundaries, could dynamically draw upon ‘in effect’, ‘applicable’ and ‘invoked’ rules suited to the granular circumstantial details of a transaction sub-event.

- It is important to emphasize that these rule dissemination capabilities in the DWDS design are not ‘socially’ unidirectional. The customer can just as easily be a rule-maker and publish their own rules over the RuleReserve network to all merchants, enabling the distributed organization of consumption. Each in their respective contexts, the platform host, the merchant, the payment services provider and the customer, can be rule-makers and rule-takers on an Internet of Rules. In all scenarios, however, it remains the rule-taker who decides what to do with any asserted obligations, permissions or encouragements that are deemed to be ‘in effect’, ‘applicable’ and ‘invoked’.

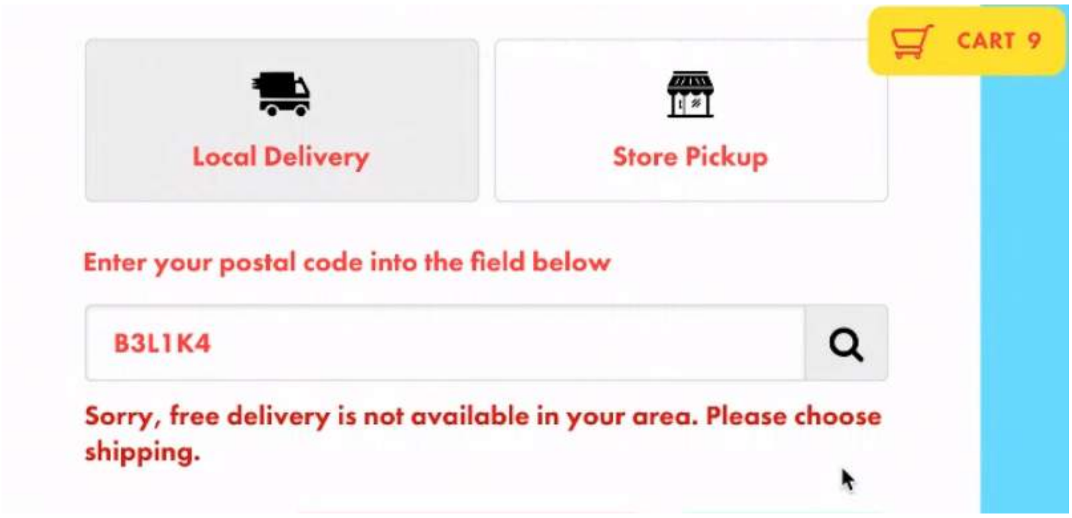


Figure 43: RuleTaker tested with a merchant’s rule based on a table of postal codes.
In this test, the hypothetical merchant configured the postal code field such that once the first three characters are entered, an [is.dwd] data package is provided to the auxiliary RuleTaker component. RuleTaker then immediately fetches any 'in effect' and 'applicable' rules from the RuleReserve network. In this illustration, the merchant’s own geographically-bounded delivery policy is among rules delivered back in the [ought1.dwd] message, and the RuleTaker component again uses the [is.dwd] package to sift out the ‘invoked’ part. The merchant has configured the shopping cart to then display a relevant response message based on the postal code.

Chapter 7: Conclusion

7.1 Purpose and Outcome

This research was undertaken to design a novel, practical method that affords any person or organization the ability to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate rules across any informatic network, with precision, simplicity, scale, speed and resilience, with deference to prerogatives, agreements and preferences. The outcome is an end-to-end design specification for 'an Internet of Rules', the components of which are being implemented by others. As such, this dissertation is 'for' project managers, and is not 'about' project management.

Chapter 1 identified a general problem of normative communication: *"Agent A, interacting with Agent B, requires knowledge of one or more externally-managed rules from Agents C..n that are 'in effect' for given contexts, are 'applicable' to a set of event categories, and are 'invoked' by particular circumstances"* amid uncertainty about agents and about rules. To solve this required the design of a general-purpose informatics solution. However this research undertaking is situated in the project management academic program of a school of administration, anticipating that it would require the conceptualization of a general framework for rule systems as a type of normative infrastructure, beyond day-to-day activity. Project ecology frames a project on multiple levels: "from the micro-level of interpersonal networks to the meso-level of intra- and inter-organizational collaboration to the macro-level of wider institutional settings. (Grabher & Ibert, 2011) (Dille & Söderlund, 2011) A *meso*-level design change that achieves widespread adoption at the *micro*-level can result in emergent and transformative *macro*-level effects for whole societies and economies. This research fills a gap in both the theory and practice of project management, concerning how any stakeholder can discover and obtain factual information about practical, logical, ethical and aesthetic rules that are 'in effect' for dates/times and prerogatives relating to identities and jurisdictions of a given context; that are 'applicable' to the class of endeavour and task being undertaken; and that are 'invoked' by a particular circumstance of the moment.

Chapter 2 on methodology opened by situating what makes a DBA (Doctor of Business Administration) different from a PhD (Doctor of Philosophy). The purpose of DBA research is to create an original way to solve a general category of real-world problems in professional practice, whereas a PhD research pursues original research of academic value to advance theoretical understanding. The methodology for the present design science research has been situated in so-called 'middle range theory', that's to say it is oriented to the functions and specifications of an intended design. Iterative experimental implementation involved artifact elaboration and iterative construction (learning through building) in collaboration with a community of scholars and practitioners. At the present point, the implemented software is at version 3.x, and with confidence it can be said that continued deployment would *enable anyone to express, publish, find and fetch rules at scale over the Internet (feasibility); across domains and use cases (generalizability); more effectively (greater outcomes) and/or more efficiently (less time/money/risk) (utility) than other existing approach.* This is assessed through reasoned analysis, not quantitative bench tests, since the operational components being implemented are still too early for useful comparative tests. The outcome can also be assessed qualitatively by the degree of alignment to a set of declared 'design virtues' and 'design norms' that are detailed in Section 2.5

Chapter 3 stepped through the philosophical foundations of how humans communicate obligation, permission, and encouragement. The most essential conceptual contribution of this work is expressed as:

$$\text{'IS + RULE} \Longrightarrow \text{OUGHT'}$$

This relation is intrinsic to an extensive of existing philosophical literature on normative and deontic logic, but it has not been previously expressed as succinctly as this. It emphasizes that rules supply the direction to human endeavour, from what 'is' and what 'ought' to be, established among two or more individuals or entities. This is the data that communicates which way is 'forward' when orienting decisions involving many people, in on-going micro-level decisions for the day-to-day management of projects, programs, portfolios or platforms, as well as in core macro-level system design initiatives, mechanisms and structures. In practical, logical, ethical and aesthetic matters, rules express obligation, permission or encouragement. MUST, MAY and SHOULD, and their various negatives and synonyms, are positioned here as a distinct class of data with direction that becomes readily discoverable and transmissible over the Internet, in a form that is directly usable by non-specialized humans and their devices, for any purpose, in any language. No previous literature in data science that I have found has categorized data which embodies intrinsic normative direction as a distinct class of data. The functional DWD Specification is limited to conveying *'normative propositions'*, and it is intrinsically '*human-centred*'.

Chapter 4 provided a comprehensive review of available approaches, focusing on core technical concepts and methods that have shaped the course of this research. This is not a feature comparison among various solutions on the market; rather it situated at the level of core concepts and methods:

- Various techniques for expressing rule logic were considered, such as a computer programming language, a graphical flow chart and many types of input/output data structure ("logic table", "truth table", "control table", "decision table", “logic gate”).
- Logic data models with binary, trinary or tetranary elements were considered, and a case was made for using tetranary logic to incorporate uncertainty to address real-word circumstances.
- Several rule processing systems were considered (rule engines, workflow processes, decision support systems, programmable logic controllers and artificial intelligence), and each was distinguished from the novel design emerging from the present research.
- The rediscovery and rehabilitation of long-forgotten antecedents from the 1950s, 1960s, 1970s and 1980s is an important part of this design research enterprise. Numerous contributions were selected from 70 years of programmable logic, on four themes:
    - Data Structuring and Transmission;
    - Tabular Declarative Programming;
    - Procedural Logic Programming and a
    - Tetranary Logic Data Model.

A lengthy and detailed **Chapter 5** explained the technical reasoning and design summary for building the IS + RULE $\Longrightarrow$ OUGHT relation into an end-to-end operational data processing pipeline. The design has been named the ‘Data With Direction Specification (DWDS)’. Once deployed, it is expected to give rise to a decentralized distributed emergent meso-level phenomenon which which has also been named “an Internet of Rules’.

The dissertation provides descriptive functional specifications for the *RuleMaker* application, the *RuleTaker* component, and the *RuleReserve* network service. Several technique were described to achieve high performance distributed and decentralized computation:

- Externalize computation from run-time;
- Transform natural language from complex to simple;
- Externalize linguistic complexity of the rule structure;
- Externalize computability by requiring rule expression to be NOT Turing-complete;
- Externalize the data processing burden with tables;
- Externalize reusable algorithms with tables; and,
- Externalize the logic.

This section then explained a novel approach for obligation, permission or encouragement to be readily discovered and communicated across the Internet. In particular the data sifting method was detailed in a series of step-by-step illustrations. There are two sifting steps. First, the RuleReserve network nodes use the reported facts to sift an initial collection of the 'In Effect' and 'Applicable' rules. Then separately, the originating RuleTaker component re-uses the same reported facts to sift from the logic gates of the pre-sifted rules, only the Output Assertions that are 'Invoked' by the facts. This whole round-trip rules-as-data process can occur almost instantaneously because there are no computationally heavy processes.

Chapter 6 documents the first end-to-end reference implementation based on the DWD Specification. This functional software was developed from the December 2022 draft of the DWD Specification (i.e. an earlier version of this thesis). A senior developer from a global commerce services firm, and a senior data scientist working full-time in a government regulatory office, programmed the core operational software components (RuleMaker, RuleReserve and RuleTaker), and demonstrated them during a public online call in March 2022. These implementations demonstrate that the design implementable and that it seems useful to some practitioners. It is beyond the scope of this dissertation to detail the substance of the various independent collaborations in which DWDS is being explored and adapted to solve practical problems. Instead, Appendix C summarizes some of the genuine external interest in the domains of cross-border trade, infrastructure finance, ecological protection, monetary anchoring, data licensing, smart contracts, algorithmic investment instruments, and regulatory technology.

In 2016 I jointly incorporated the not-for-profit Xalgorithms Foundation to manage research funding, issue contracts for supporting work, and host internal and externally-led working groups relating to DWDS implementation software and projects. Collaborations are managed through several online management applications.[79] A general-purpose Internet of Rules, first proposed through this research in 2016 (Xalgorithms, 2016) (Xalgorithms, 2021), is intended to support practical decisions and courses of action in any context where rules are 'in effect', 'applicable' and 'invoked'.

79 The Data With Direction Specification (DWDS) and various use cases are managed under Xalgorithms Foundation, of which the present author is founding Executive Director. Online activity is convened through the follow URLs:

- Websites: *https://xalgorithms.org* and *https://era.xalgorithms.org/*
- Source code management environment: *https://gitlab.com/xalgorithms-alliance*
- Project management environment: *https://xalgorithms.redminepro.net*
- Open-door community meetings: *https://meet.jit.si/xalgorithms*

The extent to which the product of this research is picked up by others outside the scope of my own management provides tangible and credible weight to validate this work. Independent implementations, as well as the decisions of institutional authors to dedicate effort in their own projects which implement or advance my design constitutes a type of peer review and validation that is, arguably, germane to a DBA style of thesis. The extent to which such results are attained validates my purpose in pursuing this work via a DBA, versus the PhD stream.

This dissertation wraps up with six substantive appendices. Appendix B outlines a unique metaphorical way of thinking about how to structure decentralized, distributed informatics systems, which shaped my approach to the DWDS design. Appendices C and D contain texts written by others. The first set consists of excerpts from industry literature about my DWDS work (sometimes referring to “an Internet of Rules”, or to the previous name “Oughtomation”, or to use cases that my design would enable). The second set of texts written by others are invited comments by three independent volunteer contributors to the first reference implementation of the DWDS design, who have been collaborating on free/libre/open licensing terms. Appendix E is an ‘Afterword’ that I prepared in December 2022, in order to review recent academic literature published in the 2019-2022 period about the current influences of informatics on the project management discipline. The final Appendix F contains excepts from a submission to a major regulatory body with suggested path for deployment. My present dissertation is a concurrent contribution to this emerging literature, but as explained at the beginning of this concluding chapter, my contribution is ‘for’ project managers, not ‘about’ project management.

7.2 Original Contributions and Useful Restorations

This dissertation describes novel conceptual and functional elements for an Internet of Rules: a decentralized and distributed method for anyone to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate normative data relating what ‘is’ with what ‘ought’ to be. This four-part system specification consists of the RuleData model, the RuleMaker application design; the RuleReserve data storage and networking design; and the RuleTaker component design.

This section lists two dozen original [O] conceptual design and methodological contributions, and a half dozen other ‘useful’ [U] restorations of existing design concepts and methods that have otherwise been overlooked or forgotten in most of informatics. The items below are referenced in the order which they appear in the dissertation. Literature references are not repeated, as they are already provided in the cited sections.

[O] Section 2.5 provides a set of ‘design virtues and norms’ which offers an original and generalizable guideline for systems development. There are many existing guidelines on system design values and ethics, but they are oriented to the behaviour of personnel (e.g. Chatila & Havens, 2019) Mine is a novel, widely applicable, and directly re-usable guideline oriented to the designed systems. Among the elements, the subsections on ‘Simplicity’ and ‘Least Power’ express these design criteria more precisely for other designers than other guides I have seen.

[U] The ‘Tabular Declarative Style’ introduced in subsection 2.5.2.6 rehabilitates a method of computing that is rarely known among current generation software developers. (My design research introduces several original [O] methods to general purpose tabular computing technique, which are listed below.) I hope this dissertation becomes a significant and useful contribution to the understanding and resurgence of tabular declarative computing.

[O] The most essential original philosophical concept contributed in this work is expressed in Section 3.1 as: 'IS + RULE ⟹ OUGHT'. This relation seems commonplace, but I have not found it expressed in a concise and explicit manner in previous philosophical literature.

[O] DWDS is grounded in the works of Wittgenstein, von Wright, Anscombe and Kalinowski in the 1950s. But in light of the shift from *norms* to *norm propositions* led by Alchourrón in the 1960s, Section 3.1 explains how DWDS differs from other functional rules systems in three ways:

- A conventional decision table requires determinacy: a single outcome. However the DWDS logic gate provides a simple way to accommodate diverse sources of uncertainty (imprecise terms; low-quality data; gradual differentiation; probabilistic states; system non-linearity). When outcomes are indeterminate, they are delegated to human agents to resolve.
- Many rules systems are designed as ‘*systems of norms’* with input condition states limited to Yes/No which need to be gapless and contradiction-free (determinate). Instead, the DWDS is a ‘*system of norm propositions’* in which input condition states include ‘Yes’, ‘No’, ‘Yes-AND-No’ as well as ‘Yes-OR-No’.
- Many rules systems limit output assertions to the elements MUST / MAY / MUST-NOT (obligation; permission; prohibition) (Von Wright, 1951) (Crawford & Ostrom, 1995). The DWDS design structures its output assertions with the elements MUST / MAY / SHOULD and their negatives (obligation-obstruction; permission-prohibition; encouragement-discouragement) (ISO/IEC, 2018) (Bradner, 1997). In Ostrom’s schema, ‘SHOULD’ is a weak form of ‘MUST’; while in DWDS there is a crucial distinction: ‘MUST’ prioritizes *rule-maker prerogative* in the social relation; ‘SHOULD’ prioritizes *rule-taker prerogative*.

[O] Section 3.1 provides an original way to distinguish deductive, procedural and normative logic, expressed respectively with LET-THEN; IF-THEN; and WHEN-THEN. “These three are commonly conflated in the published literature, since the techniques employed to resolve and to communicate all such logic problems are the same."

[O] Section 3.3 "What is an Algorithm" provides a novel linkage between two bodies of formal literature, as follows: "The computational algorithm should be understood as a precise and composable evolutionary extension of human agency. Agency is the possession of attitudinal, intellectual and tangible faculty of action to pursue a specified result. An algorithm is a method invoked by a condition to obtain a specified result and then terminate."

[O] Section 3.3 expands on this linkage with the question: "But whose agency is being extended: the algorithm user's or the algorithm designers agency?" My social view of technical algorithms is, I think, an original contribution to the theory of computing, to the theory of human agency, and to the theory of constitutional law. The latter connection is documented in Table 5 which I adapted from the work of Japanese constitutional law scholar Ken Endo (Endo, 1994), and which I also reviewed in email communication with senior Canadian constitutional lawyer Peter Hogg, author of Constitutional Law of Canada (Hogg, 2007). To avoid extending my dissertation beyond the scope of project management, I left this topic for subsequent formal journal papers. Still, my contribution during the dissertation research phase led to my being an invited leader for a session entitled: "Human-Centred Automation: Why and How", to discuss this section of my draft dissertation with a dozen academic and professional attendees in the Canadian Government's 2019 "Symposium on Algorithmic Government".

[O] Section 3.2.2 borrows a set of categories from formal linguistic theory relating to the expression of authority, commitment and gravity (Verhulst et al., 2013) in order to bring a high degree of contextual nuance to the formal theory and professional practice of rules automation. I have not encountered any other work in applied rules systems which accommodates such factors.

[O] Section 3.4.1 advances formal rules theory, as well as the field of applied rules automation, with framework for understanding and managing noise and uncertainty. I have not encountered any other work in applied rules systems which accommodates these factors.

[O] Section 3.4.2 offers an original framework to gauge qualitatively 'better' versus qualitatively 'worse' rule systems with reference to three organizational factors: level of effort; stakeholder representativeness; and logical coherence. These general system design criteria for DWDS are illustrated with an accessible and convenient metaphor that may seem contrived, but that embodies the same information thermodynamic fundamentals. As this dissertation is pursued in a Department of Administration Sciences, I will not pursue a technical digression into formal information thermodynamics (Tribus et al., 1966). However the metaphor is an invitation for others to consider subsequent whole-systems analysts through formal thermodynamics of computation (Bennett, 1982) (Landauer, 1967) (Georgescu, 2021) (Gupta et al., 2021), of network communication (Aleksic, 2013) (Shental & Kanter, 2009), and of data quality (Wright, 2016) (Gilbert et al., 2016).

[O] Section 3.4.3 resurfaces the "project ecology" conceptual framework in project management theory but then expands it conceptually by linking it with formal systems ecology theory, and then suggesting a novel *meso-micro-macro* operational sequence for meso-level rule systems.

[U] Section 4.1 is a useful comparison of diverse methods for representing a set of relations between input data and output data. The approach was adapted from an informal online source (Baker, 2004) , but was expanded to include several alternative standards (Decision Modeling Notation and RuleSpeak from the OMG, and RuleML and Notation3 from the W3C), as well as two stages in the development of the DWDS tetranary vertical I/O logic gate.

[O] Section 4.2 provides an original comparative review if three-element and four-element logic models. This includes a novel typology under what I refer to as "Tetranary Logic", grouped as "Tetralemma Logic"; "Normative & Deontic Logic"; Four-Valued Logic"; Nucleobase Logic"; and Quaternary Logic. I have not seen such a typology elsewhere in the literature.

[O] I found that the Tetralemma school of Graham Priest et al., and the Four-Valued Logic school of Neur Belnap et al. use the term "Both" very differently. In the Tetralemma school 'Both' refers to a state of being both actually True and Actually False, whereas in the Four-Valued School, 'Both' refers to a stated of being said to be True, and also said to be False. I have not seen this clarification made elsewhere in the literature.

[O] My review of four-element logic models revealed what I suggest is a confusing choice and words by the prominent school of thought in logic led by Neur Belnap. That community uses the term "None" when the term "Untold" is their intended meaning. So I suggest using {True, False, Both, Untold}. Their "{True, False, Both, None}" convention leads to confusion. I have not seen this clarification made elsewhere in the literature, rather I suspect some confusion arise due to the poor choice of word.

[O] Section 4.3.4 introduces Artificial naïvety (A∅) as a passive request-response signal mapping with declarative logic tables which retains no user data. This provides an alternative to Artificial intelligence (AI) which requires active knowledge acquisition about users and inductive reasoning. I am not aware of any other contributor to the informatics domain who describes and is designing an alternative to the AI paradigm for next-generation advanced computing.

[O] Chapter 5 describes a wholly original system design for anyone to discover and obtain factual knowledge of the rules that are ‘in effect’ for dates/times and prerogatives relating to identities and jurisdictions of a given context; that are ‘applicable’ to the class of endeavour and task being undertaken; and that are ‘invoked’ by a particular circumstance of the moment. Three sequence diagrams in Section 5.1 provided step-wise operational description of the data flows.

[O] Section 5.3.2 explains that since I was unable to find a suitable typology of controlled natural language frameworks (CNL), I developed my own, as summarized in Table 12. It has two main groups: CNL frameworks oriented to writing rules, and others oriented to analyzing rules.

[O] Section 5.3.2 also outlines a rationale for converting unstructured free-form natural language into uniformly-controlled natural language, so that it can be processed by computers over a network. The innovative tactic employed here is to introduce a “very rigid syntactic constraint” while leaving “no boundaries to semantic scope”. Whereas a Semantic Web schema *constrains semantic expression with unlimited syntactic structures*, the DWDS schema *constrains syntactic structure with unlimited semantic expression*. This is a novel way to implement a “finite state grammar” described in Noam Chomsky’s earliest work: *Syntactic Structures*. (Chomsky, 1957)

[O] Section 5.3.3 describes an entirely novel method for implementing a constrained syntactic structure described in the previous section, in order to separate linguistic complexity from the declarative sentences used to express rules. The method is illustrated with a single sample sentence in 25 languages across multiple character sets including Korean, Thai and Greek, as well as right-to-left scripts such as Arabic and Hebrew. This method is now employed directly in the RuleMaker software reference implementations.

[O] Section 5.3.5 describes the unusual mandatory constraint requiring DWDS RuleData to <u>not</u> be ‘Turing-complete’ (i.e. <u>not</u> able to be programmed to perform any function), to prioritize intrinsic security as well as system-level performance. Most other general purpose computing systems are design to ensure they are Turing-complete.

[U] Section 5.3.6 describes the uncommon design decision, although not original, to separate control data (what is) from program procedures (how to). This involves a very significant re-framing of programming from procedural IF-THEN-ELSE expressions to declarative GIVEN-WHEN-THEN expressions. Although this is not an original approach, it is referred to by Jerry and Julie Sussman as “radically different” (Abelson et al., 1984).

[U] Section 5.3.7 provides a useful classification of tabular data topologies as ‘Cartesian Product’, ‘Vertical Stack’ and ‘Horizontal Tape’. None of these is original, however no other source I have seen provided a comparison of the relative purposes of each. The DWDS uses the three topologies for different purposes.

[U] Section 5.3.8 explains an ancient method that “is so obvious that it is commonly overlooked”. Computers can look up pre-computed results from tables much faster and with less energy use than they can compute the results on-demand each time. When this ‘old school’ method is implemented on current-generation computing system, there are many circumstances in which orders of magnitude performance improvements are attainable compared with ‘conventional’ procedural data processing methods.

[O] Section 5.3.9 presents an original logic gate design that accommodates both deterministic and complex programming requirements. From a comparative review of several types of ‘truth table’ notions for binary, trinary and tetranary logic, a rationale is provided for implementing a tetranary system, using the binary digits for {0,1,2,3}, i.e. {00, 01, 10,11}. To accommodate the 'IS + RULE $\Longrightarrow$ OUGHT' data transformation, different semantic meanings are given to these values on the input and output sides: {No, Yes, Yes-AND-No, Yes-OR-No} and {NOT, MUST, MAY, SHOULD}, respectively. This enables a highly nuanced logic structure for ‘rules-as-data’.

[O] Section 5.4.1 describes three parallel representations of the same ‘rules-as-data’ for different purposes – human comprehension; data integrity and transmissibility; storage and processing efficiency. This section also presents the full ‘rules-as-data’ package in JSON expression.

[O] Section 5.4.2 describes an optional “data with direction transport protocol” (DWDTP) separated from “hypertext transfer protocol” (HTTP) for use where the DWDS would be relied upon for use cases with low very risk tolerance, such as monetary, safety, security, ecological and liberty requirements.

[O] Section 5.4.3 describes a unique identifier requirement for each rule, due to the combination of its decentralized, distributed, versioned, immutable, and time-dependent context. In order to avoid the creation of yet another identity system, DWDS creates an onion-layer method for four widely-used identifier standards to be combined, where the URI supplies the primary package. This support flexible ways to reference [rule.dwd] records, such as ‘the nearest available version’.

[U] Section 5.4.4 provides a useful design for a background diagnostic rule. Table 27 is included purely for the pedagogical purpose of having readers of this dissertation realize that even the small number of 4 values {00, 01, 10, 11} results in 256 permutations of four elements. This implicitly communicates how readily rule systems can become unwieldy.

[U] Section 5.5.1 re-purposes a distinction between two fundamentally different search methods from cognitive psychology (feature search vs conjunction search), and explains where each is being implemented in the DWDS design.

[O] Section 5.5.2 describes an original 'data sifting' method for the metadata on RuleReserve nodes. This section has no source references because I designed it from the ground up, however it will surely be similar with some earlier methods that I have not seen.

[O] Section 5.5.3 describes an adaptation of the 'data sifting' method previously mentioned, for use in the logic gates provided to RuleTaker clients. This section also has no source references due to it being designed as new process, with the same caveat that likely there are similar methods.

This particular method had many iterations. Initially I was attempting to implement the method described by Wayne Cunneyworth (Cunneyworth, 1994) for use on mainframe computing platforms, but I had to change it to suit the distributed decentralized Internet of Rules context. They changes eventually became so thorough that Cunneyworth commented that my design was more efficient and flexible than was he designed which is in use by companies such as CitiCorp, Schwabb, AmEx, VISA, AIG and other global financial companies. I first prototyped the method for sifting a tetranary vertical I/O logic gate on 26 September 2021 with some shells, sticks, stones and seed clusters in single, double and triple configurations for{0,1,2,3}, while on an small island that I canoed to in the middle of the Gatineau River.

Figure 44: A first experiment using tangible artifacts to visualize how DWDS data sifting could work (2022-09-26)

Upon returning home from the island, I formalized the method in several iterative steps. I laid out a structured version on my kitchen floor and photographed it to send to several colleagues, complete with the Input Conditions and Output Assertions labeled, along with four scenarios. All of my colleagues felt that this artisanal version was too far removed from real use cases, and generally would be too odd to include in the dissertation. So I left this part of the actual creative design trajectory un-documented in the December version of the work. Yet this is how the method of resolving DWDS logic gates did originate a half year ago, and it is exactly the step-wise procedure that is now implemented in the RuleReserve and RuleTaker software.

Figure 45: An initial structured illustration of rudimentary DWDS data sifting for informal discussion with colleagues.

The image shows a "logic gate"

- *There are three Input Conditions: "I have a stick"; "I have a shell"; "I have a little stone".*
- *There are two Output Assertions: "I have a medium stone"; "I have a large stone".*
- *There are columns of seed pods {0,1,2,3}, and these are read column by column.*

1. The first column (Scenario A) has no seed pod in the stick row, meaning "NO":"I have a stick". But the first column has a singular seed pod in the shell row, meaning "YES":"I have a shell". And so on.
2. At the top left ("[is.dwd]"), I have a shell in my hand, I have no stick and I have no stone. The only column of the logic gate that matches that eventuality is the second column (Scenario B). It can be read as follows: a triple seed pod in the stick row means "YesORNo":"I have a stick"; a singular seed pod in the shell row means "Yes": "I have a shell"; and another triple seed pod in the little stone row, means "YesORNo";"I have a little stone".
3. We can now read the Output Assertions and normative values of Column B: a double seed pod in the medium stone row, means "MAY":"I have a medium stone"; and a triple seed pod in the large stone row means "SHOULD":"I have a large stone".
4. There are two viable outcomes shown in the top right ("[ought2.dwd]"). In both cases I SHOULD have a large stone. But optionally, I MAY have a medium stone. The simple example illustrates that in this system, more than one outcome can be equally valid.

7.3 Limitations of this Research

This dissertation is limited to conceptual foundations and formal design research, with only a very superficial reach into the concurrent fieldwork that has been underway in diverse domains. In the course of this research, which has been licensed 100% free/libre/open from the outset, collaborative teams have arisen and become very active towards implementing the DWDS. A sample of external references to this design research is provided in Appendix C. However it would be impractical to incorporate all of these initiatives into this design research body of work, even in summary form. Not only do they each require considerable explanation, given the depth of change that this conceptual structure and practical implementation portends, but they are all moving forward under the initiative of other individuals in business, government and academia. DWDS and its functional elements are inherently auxiliary, making it difficult to share information here about the real project examples without having to delve too deeply into the details of the projects themselves, and in those details losing the plot that's central of the present design research. The scope and depth of this dissertation is already bordering on unwieldy, therefore a decision has been made to set the boundary to include only the theory and the design, and to including the results of some 'lab' development through working software implementation by others, but to leave the real-world implementations that are germinating in industry and government to the 'future research' category.

More generally, this research suffers from a limitation that every multi-disciplinary researcher contends with: the pursuit of multiple different domains of knowledge outside ones own core domain leaves the author at greater risk of error, than would be the case when staying safely within one's own specialization. However the class of problem addressed here does not seem to be either thinkable or solvable through any narrowly targeted domain. It's *meso*. So the narrowing of this undertaking to a manageable target has involved restricting the scope to the design of a single operational improvement in how humans communicate obligation, permission, and encouragement.

7.4 Future Research

7.4.1 Various Suggestions Received

Some new avenues of analytic research are opened by this work. At the April 2022 inaugural *'Rules-as-Data' Workshop* involving academics from ten universities (Damonte & Bazzan, 2022), some topics for future academic exploration arose in the session in which my presentation and paper about DWDS were discussed. Participants suggested that my work would be improved with:

- a typology of rule logic patterns;
- a typology for sources of uncertainly in rule expression;
- criteria for weighting the dual goals of broad rule accessibility versus formal legality; and,
- analysis of the behavioral significance of an Internet of Rules signal-to-noise ratio (i.e. the implications of improving the quality of communication between rule-makers and rule-takers).

The first of these topics for future exploration identified in the spring of 2022 expanded as a result of my conversations with Magdalena Pradilla Rueda at the end of the same year. My Section 4.2 "Alternative Logic Data Models" has improved, but it requires more rigorous attention to the different philosophical premises underlying the various logic schemes, and would benefit from a formal metalogic scaffold. The last topic in that list is related to questions that arose at the very beginning of my research trajectory in project management theory:

1. Must we make a blanket assumption that all projects move us ‘forward’? Might one reasonably distinguish a ‘forward’ project versus a ‘backward’ project?
2. How do the principles of human-centred automation affect the ability of project managers to assert their prerogative to manage the projects which they are responsible for, versus getting displaced by managers of informatic systems they depend upon? ?
3. What methods and metrics would be helpful to undertake a genuine comparative performance assessment and refinement of rules, sets of rules, and whole rule systems? [80]
4. What potential is there for ‘*meso-level projects*’ to advance rule systems infrastructure performance to tackle the most pivotal issues of our time?

The fourth question is being pursued already, as mentioned above, in relation to cross-border trade, infrastructure finance, ecological protection, monetary anchoring, data licensing, smart contracts algorithmic investment instruments, and regulatory technology (Appendix C). The meso-level project concept is not new, but the project management literature of academia has not previously internalized the *mico-meso-macro* framework that Kurt Dopfer, John Foster and Jason Potts tabled two decades ago (Dopfer et al., 2004). They distinguish the methods and dynamics of *micro-level* projects that engage decision-makers amongst organizations, *meso-level* projects that seek to advance normative infrastructure and practices amongst industries and markets, and *macro-level* projects that are designed to influence characteristics of whole societies and economies. DWDS use-case implementations are meso-level projects.

7.4.2 Ongoing Technical Methods Research

In parallel to the project management domain, there are interesting and useful technical research questions to be resolved as work proceeds on the reference implementations of RuleMaker and RuleTaker applications and the RuleReserve network

The software and network reference implementations provide a basis for a wide range of interesting and valuable design research. Some specific questions being pursued with colleagues are:

- What exact methods will be optimal for forward and backward chaining of [rule.dwd] records, and for calling data from [lookup.dwd] tables?
- What types of arcs may be created among SubsetRuleReserve nodes, and what types of SubsetRuleReserve nodes might there be (general access; constained access)?

80 The three DWDS Postulates given earlier in Section 3.4.2 are:

- An optimal ***rule system*** within a jurisdictional cluster of arbitrary individuals and entities is one that demands the least effort for them to categorize and communicate their respective normative propositions.
- An optimal ***rule*** between any two individuals or entities is one that is centred upon their respective priorities, while also intersecting their shared points of agreement.
- An optimal ***set of rules*** among multiple individuals or entities is one which all the rules together reveal an emergent straight line of reasoning.

- Generally what sorts of programming problems are addressed more efficiently and effectively with tabular declarative versus procedural imperative methods?
- How should large multi-factor data streams (e.g. selected satellite data, market data, epidemiological data) be structured into [lookup.dwd] records in a manner that will be easy and efficient for [rule.dwd] sentences to draw upon?
- What is a practical way to incorporate [rule.dwd] and [lookup.dwd] records into agent-based modeling environments, in order to test rules prior to real-world deployment?
- What graphical symbols and colours to display NOT, MUST, MAY and SHOULD normative semantics would make the most intuitive sense across cultures, for implementation in the RuleMaker user interface?

7.4.3 Improved Computational Linguistics

Section 5.3.4 considered the utility of reconciling DWDS logic gates with the existing RuleSpeak standard. This would facilitate the adaptation of a large volume of conventional natural language rules into controlled natural language sentences and logic gates of the DWDS.

Widespread transcription for deployment of an Internet of Rules would ideally proliferate through decentralized self-organization. When the Deputy Director for digital governance in a national government asked me in early 2021 how this might be accomplished, I described the following programme in a reply email (lightly edited from my original message):

> “I presume that the purpose of your question is to consider whether this approach can scale to tens or hundreds of millions of rules that have to be made computationally operational.
>
> We have considered some ways to create the *human* equivalent of "massive parallel processing" to get there. I'll explain...
>
> Each year in your country about 1,500 law graduates pass the National Bar Examination. Imagine if your organization were to invite all the country’s law schools to incorporate into their programs a type of capstone project which involves expressing one statute in JSON with the most up-to-date DWDS RuleData specification. Longer or more complicated legislation could involve larger student teams. The volunteering students, their professors, teaching assistants and external advisors (i.e. working lawyers specialized in the relevant sub-domain) would make the effort to ensure that the nuanced original legal meanings are accurately conveyed in the sets of declarative statements in RuleFiniteStateGrammar. Your organization could also offer recognition and/or bursaries for the best results. One of the criteria would be whether the output patterns and lexicon produced by a team have become useful towards simplifying or to improving the quality of work by other teams. All of this work would be free/libre/open licensed. So one of the essential criterion would be whether a team has made efficient and effective use of patterns and lexicons produced by other teams. This gamification design rewards co-opetition.

All the results would be empirically testable in three ways:

1. Invite private companies to test the resulting JSON-expressed DWDS rules in their existing systems;

2. Test the rules with transactions/events that are generated in an economic agent-based model.

3. Test rule discovery across an "Internet of Rules" created by a network of DWDS RuleReserve nodes. (Our three XRR reference implementations are still being built, but there's no requirement that any party has to use our components end-to-end. Independent implementations of RuleReserve can be created by anyone with relevant understanding and know-how.)

Similar initiatives can be run in other jurisdictions. This approach works equivalently in any language. Each jurisdiction can have a winning team.

Then, some international competitions can be run, where winning teams compete to express clauses from a highlighted multilingual international law or agreement. I imagine a 3-day competition which begins with the contestants not knowing which international legal text will be the focus (trade law; mergers & acquisitions law; securities law; aviation rules; etc. At the starting bell, all teams are pointed to the current official site for the target agreement, with whatever format it currently has. At the end of the three days the winning team will be the one that has the best quantitative and qualitative result. Initially it might seem that a default incentive would be to target the easiest clauses in order to transcribe the highest number of clauses, but because of the co-opetition criteria, there is also an incentive to target the most difficult clauses in order to produce elegant patterns and lexicon improvements that will be adopted by competing teams to save time in order that they can produce more clauses.

Teams should be able to use any free/libre/open source machine parsing systems that they want to use: their choice of system can be secret until the end of each competition. This would also stimulate an enormous push for better machine interpretation.

I wonder if my country's team can beat your national team in the first season? Core contributors like me cannot also be on competing teams." (Potvin, email, January 29, 2021)

The above style of open collaborative research undertaking would create an opportunity for specialists in legal linguistics to explore how conventional natural language rule expression in legislation, contracts, and other document types can be structured for concurrent human-readable, machine operable and transmissible, and legally accurate expression, combining RuleFiniteStateGrammar syntactic structure with the RuleSpeak general semantic standard and various Semantic Web standards.

7.4.4 Common Sense-Making Through a Period of "Incommensurable Paradigms"

The 'Data With Direction Specification' was being finalized just as the 2020s sheared every community (on Earth, it seems) into two or more incommensurable paradigms.

In *The Structure of Scientific Revolutions*, the philosopher-advocate of 'a normal science' Thomas Kuhn described a paradigm as being a shared cognitive framework of interpretation, explanation, validation and expectation that affords a society a degree of underlying consensus about structures, processes, know-how and rules that guide behaviour and shape the future. (T. S. Kuhn, 1962) Kuhn was concerned with periods of disorder and confusion among irreconcilable paradigms:

> "Like the choice between competing political institutions, that between competing paradigms proves to be a choice between incompatible modes of community life. ... When paradigms enter, as they must, into a debate about paradigm choice, their role is necessarily circular. Each group uses its own paradigm to argue in that paradigm's defence. ... Yet, whatever its force, the status of the circular argument is only that of persuasion. It cannot be made logically or even probabilistically compelling for those who refuse to step into the circle. The premises and values shared by the two parties to a debate over paradigms are not sufficiently extensive for that." (Kuhn, 1962, p. 94)

His description of the resulting social dissonance is likely familiar to anyone in the 2020s:

> "To the extent, as significant as it is incomplete, that two scientific schools disagree about what is a problem and what a solution, they will inevitably talk through each other when debating the relative merits of their respective paradigms. In the partially circular arguments that regularly result, each paradigm will be shown to satisfy more or less the criteria that it dictates for itself and to fall short of a few of those dictated by its opponent. (Kuhn, 1962, p. 109)

The philosopher-advocate of a 'competitive pluralistic science' Paul Feyerabend argues that Kuhn's *Structure of Scientific Revolutions* describes a process without direction: "He has failed to discuss the aim of science" (Feyerabend, 1970, p. 201). Feyerabend bluntly illustrates the problem:

> "Every statement which Kuhn makes about normal science remains true when we replace 'normal science' by 'organized crime'; and every statement he has written about the 'individual scientist' applies with equal force to, say, the individual safe-breaker. ... He knows that he will receive the more money and rise the faster on the professional ladder the better he is as a puzzle-solver and the better he fits into the criminal community. Money is his aim. What is the aim of the scientist? And, considering this aim, is normal science going to lead up to it? Or are perhaps scientists (and Oxford philosophers) less rational than crooks in that they 'are doing what they are doing' without regard to an aim? (Feyerabend, 1970, p. 200)

In Kuhn's model of paradigms, rules are methodological instructions for solving puzzles. Kuhn's view of science builds in no role for normative direction in general, nor for deontic virtue in particular. Although it may be rigorous, it is aimless.

Today in the 2020s people finding themselves in incommensurate paradigms relative to their colleagues and communities are nevertheless faced with negotiating on-going *micro-level* decisions for the day-to-day management of projects, programs, portfolios and platforms, as well as managing the core *macro-level* system infrastructures, supply chains, essential services and ecosystems that everyone depends upon.[81]

In 1970, the philosophical opponents Kuhn and Feyeraband agreed that a *novel* language—not implying a *'neutral'* language—would be an essential preliminary step:

81 John Sydenham Furnivall's timeless description of a 'plural society' applies here: "they mix but they do not combine". (Furnivall, 1948, p. 304) The "two or more groups live side by side but separately within the same political unit", however he makes the crucial observation that they still mix in the market place, in buying and selling. Furnivall refers to this as a 'plural economy'. (Furnivall, 1945, p.168) (Potvin, 1986)

> “The point-by-point comparison of two successive theories demands a language into which at least the empirical consequences of both can be translated without loss or change. ... Ideally the primitive vocabulary of such a language would consist of pure sense-datum terms plus syntactic connectives. ... Feyerabend and I have argued at length that no such vocabulary is available. In the transition from one theory to the next words change their meanings or conditions of applicability in subtle ways. ... What the participants in a communication breakdown have then found is, of course, a way to translate each other’s theory into his own language and simultaneously to describe the world in which that theory or language applies. Without at least preliminary steps in that direction. ... In the absence of a neutral language, the choice of a new theory is a decision to adopt a different native language and to deploy it in a correspondingly different world. (Kuhn, 1970, p. 277)

This is where the pragmaticism emphasized in methodology Section 2.1 now has equipped us to utilize the foregoing theoretical point as a pivot to solve in practice a general class of real-world problem. The Data With Direction Specification (DWDS) provides a general-purpose *meso-level* infrastructure (an ‘Internet of Rules’) to support communication about which way is 'forward' through this period of disorder and confusion. The DWDS RuleData, operationalized in the RuleMaker application, is designed as a general-purpose utility with a very basic constrained syntactic structure that is flexible to configure, with unlimited semantic range, and tolerance of vernacular, to enable the transmission of information related to any domain, that can be concurrently expressed in multiple languages, and that can reflect multiples paradigm. Footnote 48 on page 166 relates a conversation about a community that had no way to express something that most people in the world take for granted, and still we were able to proceed unhindered, with approximate functional equivalency.

Amid the clash of paradigms that surfaced in the early 2020s, there is a critical need for participatory research for common sense-making. Meta-rules described in Section 3.3.2 may be particular to a paradigm as this is meant by Kuhn. But just as people of completely different ideologies look in the same direction to a sunrise, and prefer shelter from the rain, Section 3.1 of this dissertation presented a rule as any practical, logical, ethical and aesthetic directional relation communicated among two or more people to associate what ‘is’ and what ‘ought’ to be, : 'IS + RULE ⟹ OUGHT'. This pluralistic approach to normative expression, tolerant of interpretation, opposition and some degree of non-conformance, can enable the negotiation of common sense across incommensurable paradigms.

Appendix A: Thesis Project Timeline

This doctoral design research was initiated in 2013, took shape operationally in 2015, attracted multiple genuine implementation communities beginning in 2016, and achieved operation design completion in 2021. All of 2022 involved critical review and refinement of design rationale and its explanation, including incorporation of numerous improvements realized in the course of community efforts on the reference implementations of RuleMaker, RuleReserve and RuleTaker.

The foundational research, design and preliminary implementation of the software components of an Internet of Rules was largely completed from Q1/2016 through Q4/2021, and various related unpublished exploratory partial drafts of the present dissertation were are also prepared during that period. The section linking abstract information theory to this topic I originally researched and wrote while enrolled in a doctoral program at University of Waterloo. [82]

The operational proof-of-concept implementation of the Internet of Rules software, on production-class platforms, achieved alpha-testing status in Q4/2018, with a small number external researchers participating on 100% free/libre/open licensing terms.

82 I did not complete my 1992 PhD program due to a University of Waterloo policy which was interpreted by administrative staff at that time (contrary to the views of my academic supervisors, James J. Kay and the department Chair) to prohibit me from receiving funds under a professional services research contract that The World Bank issued for me to undertake that systems design research, versus a conventional academic grant. None of the granting bodies that I was aware of had a category for the design research work I was undertaking at that time.

My report on that original research, entitled *"Classification and Appraisal Criteria for Conservation Investments: A proposed general framework" (Potvin, 1992)* was shortly thereafter quoted in the book: *"Key Concepts and Terminology of Sustainable Development"* by Mohan Munasinghe and Jeffrey McNeely, issued by The Word Bank for the United Nations University (Munasinghe & McNeely, 1995, p. 27). The references to my early work in their book are reproduced below. (In the excerpt from my report, which they quoted, I used the technical term *'exergy'*, which is an energy gradient (difference), however they mis-quoted me by using the term *'energy'* which is a total.)

> "As scientists discover more complex interconnections in biophysical systems, which in turn underlie the productive basis of human society, then the preservation of whole ecosystems (natural capital) may be viewed in economic terms. *An extreme version of this approach is centered around the so-called Gaia hypothesis, which states that the totality of life on Earth is responsible for controlling the temperature, chemical composition, oxidizing ability, and acidity of the Earth's atmosphere (Potvin 1992)*. ... Economic activity that imposes unsustainable levels of stress on the natural environment may generate negative feedback effects. *Using reasoning somewhat similar to the neoclassical argument for the substitutability of capital, Potvin (1992) states that, "at the end of a period during which depletable inventories are drawn down for use, if structural and chemical energy (sic)[exergy] newly embodied in things of human design and manufacture exceeds the energy (sic)[exergy] lost to reserves themselves, then exploitation of these inventories is consistent with ecosystem self-organization."* This enables, for example, humanity to use a finite mineral and fossil inventory in order to generate a perpetual stream of income, assuming that society invested all the rents from the resource. If capital goods are acquired rapidly enough to make up for the continually declining use of resources, resources can continue to be used and reserves remain positive. A vital element of the Cobb-Douglas production function, Q = K-R'-, is that each input is essential. If either capital stocks or resources are run down to zero, then output is zero (Hartwick and Olewiler 1986). (Munasinghe & McNeely, 1995, 26, 27)

Research Calendar

Months	Objective
December 2019	Research proposal submitted to jury (accepted)
October 2021	Dissertation submitted to first jury (2022-10-22)
March 2022	Jury written comments received (seeking major changes)
June 2022	Expanded dissertation submitted to second jury (2022-06-20)
October 2022	Jury written comments (seeking minor changes)
December 2022	Expanded dissertation submitted to second jury (2022-12-)
January 2023	Public defense (2023-01-12)
February 2023	Adjusted dissertation submitted to second jury (2023-02-06)
March 2023	Adjusted dissertation in final version deposited with Dean’s office

Continued experimental development of the working systems and associated documentation will out-live the academic research schedule of this doctoral undertaking.

Appendix B: Geometry of a Bubble Cluster as a Design Metaphor

The purpose of this appendix is to explain a metaphor of a bubble cluster which, although apparently unrelated to the subject of this thesis, was employed by this system designer to simplify his thinking about a system of relationships that is otherwise difficult to visualize. For more than twenty years I have found this to be a helpful mnemonic device when designing and managing distributed decentralized systems.

The function of metaphor in theory-building was formally examined in 2013 by Cyril Foropon and Ron McLachlin (Foropon & McLachlin, 2013). It is a common technique in both formal and popular communication because metaphor can supply a conceptual scaffold for sets of ideas (Bartlett, 1932a)(Rosch & Lloyd, 1978).

The soap-bubble then may be used to give a numerical solution of an optical and of an electrical problem.

Plateau gives one other geometrical illustration, the proof of which, however, is rather long and difficult, but which is so elegant that I cannot refrain from at least stating it. When three bubbles are in contact with one another, as shown in Figure 5, there are of course three interfaces meeting one another, as well as the three bubbles all at angles of 120°.

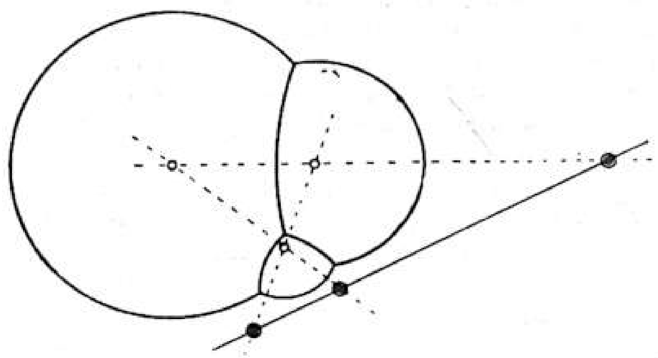


FIGURE 5

The centres of curvature also of the three bubbles and of the three interfaces, also necessarily lie in a plane, but what is not evident and yet is true is that the centres of curvature, marked by small double circles of the three interfaces, lie in a straight line. If any of you are adepts in geometry, whether Euclidean or analytical, this will be a nice problem for you to solve, as also that the surface of the three bubbles and of the three interfaces is the least possible that will confine and separate the three quantities of air. The proof that the three films drawn according to the construction of Figure 3 have the curvatures stated is much more easy, and I should recommend you to start on this first. If you want a clue, draw a line from the point where the dotted line cuts *cg* parallel to *cf*, and then consider what is before you.

Figure 46: Excerpt from: Boys, V. (1931). The Soap Bubble. In J. Newman (Ed.), 1956. The World of Mathematics (Vol. 2, pp. 891–900). Simon & Schuster.

My reflections on dynamically optimizing geometry of bubble clusters as a metaphor during DWDS system design are rooted in the field of information thermodynamics to a degree that is beyond the scope of the present dissertation,[83] I trust that a brief explanation will suffice. For as Xiaohu Ge and Litao Ya observe in a 2022 paper, all real systems, including those designed for information processing and transmission, are subject to the first and second laws of thermodynamics:

> "Since all information communication systems are implemented by physical devices and all physical devices must be obeyed by basic physical theorems, e.g., thermodynamics theorems, all information communication systems must be governed by basic physical theorems. Consequently, information communication systems could be regarded as a type of thermodynamic systems coupled by the system status changing and the energy dissipating. ... When the information communication system is defined as an open and evolutional thermodynamic system, the optimal information communication solution could be developed by adaptively matching energy with different information communication status and processes. Hence, we propose to design the information communication systems based on thermodynamics theorems." (Xiaohu & Litao, 2022, p. 4)

Similarly in a 2018 PhD thesis, Alexander Blades Boyd explained that the pursuit of "thermodynamically efficient computing" involves perceiving every input/output information processor as a sort of "information transducer" designed to use *high quality physical materials, energy, and stored actionable-information* to convert input *low quality information-as-surprise* into an output as additional *high quality additional stored actionable-information*. (Boyd, 2018, p. 3, 49, 90-93) There is a large body of related theoretical and empirical work based on the second law concept 'exergy'. (Aleksic, 2013) (Gupta et al., 2021)

The DWDS operational design in general, and the "Three Postulates for Optimal Rule Transmission Systems" in particular, as stated in Section 3.4.2, were shaped by conscious reflection upon thermodynamic optimization in the design of an efficient information processing and transmission system. There is an intrinsic common thread in thermodynamic optimality across different types of systems, but there should be no need to go deeper here than.

Any number of bubbles could be illustrated, but we'll use a minimal three-part diagram described by Joseph Plateau a century and a half ago (Plateau, 1873) (Harrison, 2014) (Harrison & Pugh, 2015). The illustration here is redrawn from "The Soap Bubble" by British physicist Sir Charles Vernon Boys of the Royal College of Science (now Imperial College London), shown in Figure 46. (Boys, 1956)

Figure 47 portrays a simple set of entities and relationships. The style of diagram on the right is the more common way to represent multi-entity collaboration, in the tradition of Leonhard Euler (Euler, 1768), John Venn (Venn, 1880) and Charles Peirce (Peirce & Marquand, 1883) (Peirce, 1885) but in applied management the areas should not overlap. In the drawing on the left, using the geometry of a cluster of bubbles, each entity exists autonomously, related without overlapping. (Moktefi & Shin, 2012)

83 It is relevant that my Section 3.4.1 "Signal and Noise in Rule Transmission Systems: Insights from Information Theory" is situated just ahead of Section 3.4.2 "Three Postulates for Optimal Rule Transmission Systems".

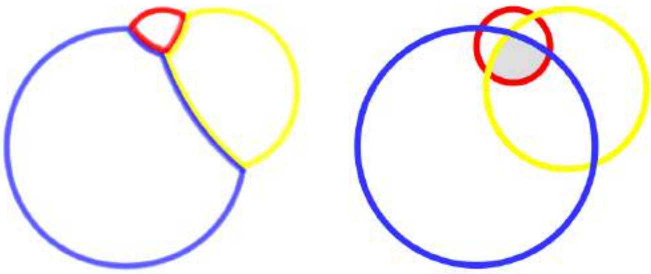


Figure 47: Bubble Cluster Interface Diagram Versus the Euler-Venn-Peirce Overlapping Areas Diagram

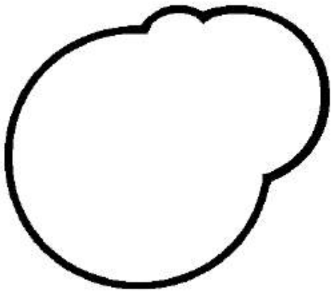


Figure 48: Each Jurisdiction has a Conceptual Boundary (Macro-Level)

A *macro-level* jurisdiction is portrayed in Figure 48 with the outline of a cluster among several parties pursuing a shared set of normative assertions. The utility of this representation for the *micro-level* and *meso-level*, is to consider them in the context of a *macro-level* identity.

In Figure 49 we assign the centre-point of each bubble to represent a *micro-level* constituent individual or entity within the jurisdiction. The 'dots' are used to represent entities, and lines are used to represent boundaries. Each constituent retains an area of autonomous agency, so that each dot is illustrated as separate from, but interacting with the other entities. Any cluster of autonomous relationships involves multiple paired interfaces. In real relationships, as well as in this geometrical metaphor, interactions may be contingent but they are not random.

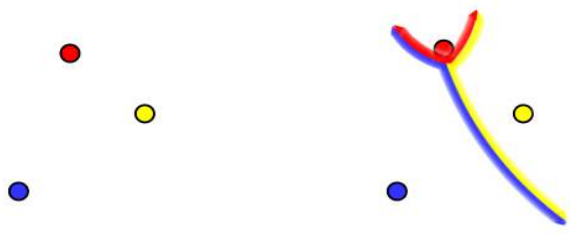
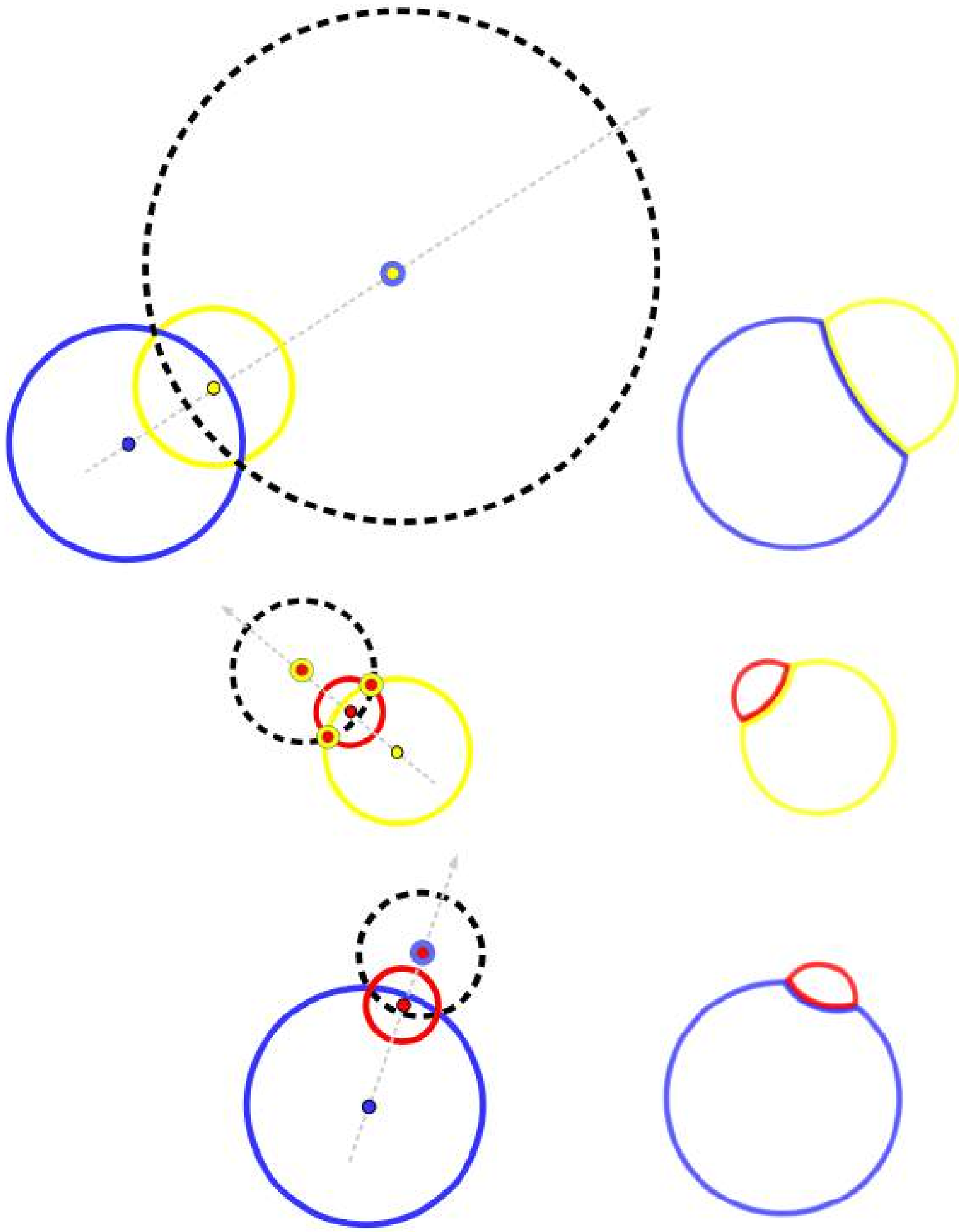


Figure 49: Groups of Three or More Entities Also Have Paired Relationship Interfaces(Micro-Level). Here 'dots' are used to represent entities, and lines are used to represent boundaries. The reason for this particular arbitrary configuration will become apparent in Figure 51.

The interfaces in this system are shaped by *meso-level* rules, in the sense described by Kurt Dopfer (Dopfer et al., 2004). Plateau identified the physical forces which determine the geometrical shapes of interfaces between bubbles in clusters. He demonstrated that a cluster of bubbles of arbitrary size which are small enough to hold their spherical stability will shape themselves to the least possible surface area required to confine and separate the given quantities of air within the system (Plateau, 1873). This leads to the first reflection: ***An optimal rule system within a jurisdictional cluster of arbitrary individuals and entities is one that demands the least effort for them to categorize and communicate their respective normative propositions.***

As illustrated in Figure 50, Plateau found that the shape of the interfaces between any two bubbles aligns to the arc of a circle (or sphere) that is itself centred on a straight line extending through their two centres. The arc intersects the shared contact points located between the two bubbles. From this arises the second reflection: ***An optimal rule* *between any two individuals or entities is one that is centred upon their respective priorities, while also intersecting their shared points of agreement.***

Figure 50: Interfaces Between Bubbles are Shaped in a Particular Way (Meso Level)



When multiple bubbles are clustered, as in Figure 51, Plateau also found that the interfaces formed as the arcs of the circles (spheres) to which their paired interfaces align, reveal that they are all centred upon an emergent straight line. And from this arises the third reflection: ***<u>An optimal set of rules</u> among multiple individuals or entities is one which all the rules together reveal an emergent straight line of reasoning.***

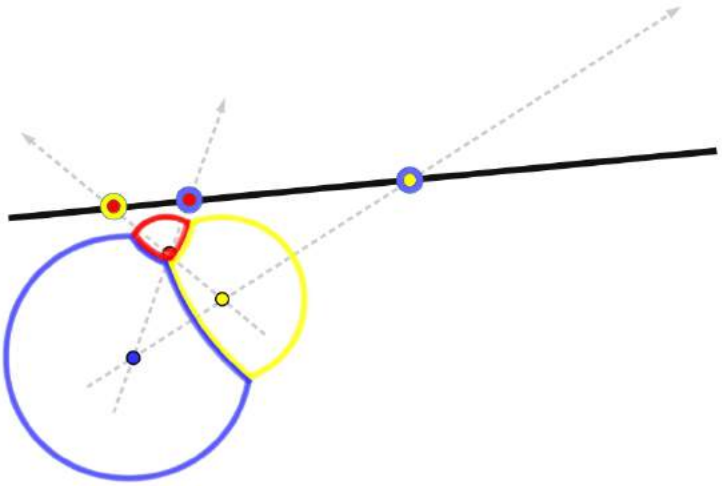


Figure 51: Entire Bubble Clusters Demonstrate an Emergent Alignment (Meso Level)

Appendix C: External References to This Research

The many developmental drafts of this dissertation were shared under the Creative Commons Attribution 4.0 International License, and its software reference implementations were shared under the Apache 2.0 and GNU Affero General Public License 3.0 in order to facilitate on-going peer review and guidance, and to thereby optimize for the feasibility, generalizability and utility of this design outcome.

Following is a brief selection of excerpts[84] from other parties to the potential use of the results of my design research in various sectors. Some of these refer to Xalgorithms Foundation, the federally-incorporated (est. 2015) not-for-profit for which I am Executive Director. It was created to effectively manage research funding, contracts for supporting work, and working group communications, relating to the substance of this dissertation project.

Some of these sources refer to ‘Oughtomation’. Recently I renamed this to the ‘Data With Direction Specification (DWDS)’, as it appears in this final version of this dissertation.

Potential Use of DWDS for Cross-Border Trade Facilitation

> **Mohun, J., & Roberts, A. (2020).** ***Cracking the Code: Rule-making for humans and machines*** **(OECD Working Papers on Public Governance No. 42; OECD Working Papers on Public Governance, Vol. 42). pp. 48-49.** (Mohun & Roberts, 2020) *https://doi.org/10.1787/3afe6ba5-en*
>
> *This report from the Organization for Economic Cooperation and Development is primarily focused on a “rules-as-code” approach (discussed in Section 4.1), which involves platform-specific programming. My platform-independent “rules-as-data” approach is highlighted in a separate box.*
>
> *“Box 4.4. Automating international trade rules.*
>
> All parties require simultaneous, consistent and verifiable data about their obligations regarding inter alia licensing, tariffs and taxes. An ‘Internet of Rules’ (IoR) (Potvin et al. 2020) – a networked repository of executable commercial policies – could fulfill the necessary cross-platform function. ...
>
> Members of the not-for-profit Xalgorithms Foundation have designed an open source method and general-purpose online service that enables any organization or individual to publish, discover, fetch and prioritize rules in the form of JSON control tables. This is done in a simple tabular style that is readable by non-technical people and directly usable by computers for data filtering and transformation. Once any rule or reference table is expressed in this ‘rules-as-data’ form, it can be directly exchanged among, or embedded into, any application built in any programming language and either used natively or auto-transcribed into ‘rules as code’ form. This promises to create, in essence, an 'Internet of Rules'. The design is usable with both ‘single window’ and 'distributed' architectures for trade, commerce, logistics and value-chain administration and is compatible with a diversity of other use cases. Trade, fiscal, and related statutes can include an attached 'schedule' with control table(s), or they can employ an ‘incorporation by reference’ clause to authoritative online sources where they are maintained.

84 These quoted excerpts from third parties explain the concepts *through substantial use of my own orginal texts* which are all under the Creative Commons Attribution 4.0 International License. In relation to copyright law, my quoting them at length is a correct use of a third party derivative expression of my original expressions of these domain applications.

Central to realizing the benefits of this approach is interoperability. That is, it must be possible for various actors and enterprises, of any jurisdiction, to discover, access, inspect and run the rules. This use case further highlights the role for technical standards. Rapidly changing trade policies could be more easily published, maintained, used and tested via a common Internet methodology that works with all applications. Ensuring that this ‘rules as data’ method of expression is human-readable also helps to ensure validation of the integrity of automated taxes, exemptions, credits, and import/export duties.

In Atkinson’s assessment, the development of an interoperable, accessible and consumable IoR could usher in a new era of trade or ‘Trade 3.0’ where ‘the distinctive character…is that countries will be able to publish both natural language and digitally executable language versions of laws and regulations’. This has the potential to democratize access to international trade in ways that are specific and directly quantifiable. For example, governments would be able to ‘see’ real-time market responses to rule updates, ascertained from signal-generation or automated reporting in a rule (within all appropriate disclosure controls).”

Government of Chile. (2021). Inception of ‘an Internet of Rules’ in the domain of international trade. Chile launches a pilot to simplify access to and use of international trade rules and data via the Internet. Media Communiqué [was] to be released at the 12th World Trade Organization (WTO) Ministerial Conference (MC12), Geneva (1 December 2021). Undersecretary of International Economic Relations (SUBREI), Ministry of Foreign Affairs of Chile.

This was postponed, and the following note is provided on the WTO website: “In April 2021, members agreed that MC12 would take place in Geneva from 30 November to 3 December. However, an outbreak of a new highly transmissible strain of the COVID-19 virus and resulting travel restrictions led to a General Council decision on 26 November 2021 to postpone MC12 indefinitely.” Then, national elections in Chile in December 2021 led to a change of government, resulting in personnel changes, with the result that the collaboration referenced here has been delayed. The present authorI remains in communication with government and university personnel in Chile in relation to this initiative, now including staff of the Chilean National Library of Congress.
https://www.wto.org/english/thewto_e/minist_e/mc12_e/mc12_e.htm

GENEVA, SWITZERLAND - 1 December, 2021

On the margins of the twelfth World Trade Organization (WTO) Ministerial Conference (MC12) in Geneva, the Undersecretary of International Economic Relations (SUBREI), of the Ministry of Foreign Affairs of Chile has launched a pilot programme to collaborate in creating a free, global, and open system of trade rules in digital form. The design makes commercial rules easier for people to understand in natural languages as well as more efficient for computer automation across a variety of trade processes, including tariff administration, regardless of platform.

Chilean stakeholders will be contributing technical know-how and conceptual design, building upon the country's long tradition of informatics theory and practice. SUBREI, on behalf of the Government of Chile, will participate in Xalgorithms Alliance (XA) and will facilitate the involvement of Chilean academic institutions and students. The first school to join the pilot is the Program of Law, Science and Technology of the Catholic University of Chile (UC).

Following a successful pre-pilot workshop in October 2020, stakeholders agreed to collaborate with Xalgorithms Foundation through ‘free / libre / open’ methods and licensing.

“We created a simple and direct way to express obligation, permission and encouragement as a distinct class of data”, said Joseph Potvin, Executive Director of Xalgorithms Foundation.

“The Data With Direction Specification is designed to enhance the capability of managers to manage”, he said. “Basically, rules-as-data would reduce barriers to human comprehension of contracts and laws, and at the same time simplify their automation.”

Xalgorithms Foundation joins with the Government of Chile’s digital economy team in inviting other public sector, commercial, academic, and civil society participants to help bring forward this decentralized and distributed ‘Internet of Rules'.

The Chilean Government sees new potential for the integration of trade with stewardship, seeking more coherent inter-jurisdictional governance, as well as more efficient project management throughout value chains.

This method of expressing rules was developed largely by Potvin as the focus of his doctoral dissertation at University of Quebec (UQO) in Canada, with research funding and tabular programming know-how from Ottawa-based DataKinetics, along with thoughtful input from a wider community of free/libre/open contributors.

It will now be adapted and extended by the Government of Chile to implement digital law at-scale. The Chilean team seeks to be the first jurisdiction to publish trade rules in a standardizable data package that can be easily used within any application on any platform.

The wider goal is to foster the incremental emergence of a distributed general-purpose “Internet of Rules”.

“This public digital infrastructure aims to provide a simpler way for both humans and machines to interface with rule systems, which over time have become too complicated. The simpler Internet of Rules approach will be especially helpful for SMEs, yet will help enterprises of any level by greatly reducing the effort needed to access global markets,” said Chile’s Vice Minister for trade, Rodrigo Yañez.

The Undersecretary of International Economic Relations (SUBREI) is a public entity, dependent on the Ministry of Foreign Relations, whose purpose is to execute and coordinate the Government's policy in the field of International Economic Relations; the representation of Chile's interests in multilateral forums; and the negotiation and implementation of the network of free trade agreements, among other matters.

Xalgorithms Foundation Inc. is a not-for-profit corporation whose purpose is to provide services to the Xalgorithms Alliance, and its participants to collaborate to create and maintain the functional free/libre/open source components for an Internet of Rules: RuleMaker, RuleTaker, RuleReserve and RuleSchema.

Atkinson, C., & Schubert, N. (2021). Augmenting MSME Participation in Trade with Policy Digitalization Efforts. *Trade, Law and Development*, *13*(1). (Atkinson & Schubert, 2021) *http://www.tradelawdevelopment.com/index.php/tld/article/view/9*

“Akin to the system of international economic governance between countries, the Internet is a set of networks and software elements that allows for exchange through protocols and standards. To extend the functionality of the Internet for the standardized transmission of rules, a general- purpose computational method known as ‘oughtomation’ is now emerging. Joseph Potvin has put forward a design: [Data With Direction (DWDS)][85] represents a different sort of pursuit: a general- purpose method to communicate rules as simple data, with minimal dependencies, equivalently usable by any application, in any language, on any device, without retrofits or refactoring. A peer-to-peer decentralized network instead lets each autonomous node accommodate the [Application Programming Interfaces] (APIs) of the applications operated by end-users, so that the end- user controls this, and lets the default for each node function in both ‘server’ and ‘client’ roles. [DWDS]’s tabular ‘Rules as Data’ [RaD] formats are optimized for efficient storage, querying and computation.

Providing functional inputs for any system, the oughtomation method avoids many complications of the RaC approach to initially express rules in a ‘code’ form (i.e., siloed into a given procedural programming language). In contrast, oughtomation presents a way to fully express rules as platform-agnostic Java Script Object Notation (JSON) data packages. The method gives effect, “to must, may and should assertions amongst individual and organizational agents to … use upcoming action data to filter rules on the Internet, map the data to input/output tables in the rules … and then, determine out how the action must, could or should be carried out.”

85 This published article used the earlier name for DWDS, which was ‘Oughtomation’. It is changed here for clarity.

Thus, an IoR, “is created when computational algorithms can be readily transmitted from any independent source repositories within which they are maintained, to any applications that would use them.” This infrastructure can enable computational rules, system integration, and process automation on a global scale as well as accommodate the dynamic nature of the Internet and changes in laws and regulations. In complement to the method, Xalgorithms Foundation is concurrently developing essential software components as reference implementations under ‘free/libre/open’ licenses, so that any person or entity can use, adapt or re-implement them to operationalize normative RaD online.

Footnote [in the original article]: “An Internet of Rules” was coined in 2016 by Joseph Potvin, Executive Director, Xalgorithms Foundation, to describe the emergent system enabled by [DWDS]. This is elaborated in his forthcoming doctoral thesis at University of Quebec, Joseph Potvin et al., (Forthcoming, 2021, Unpublished) Data With Direction. Dissertation in the Dept of Administration (Project Management), Université du Québec).

Atkinson, C., & Potvin, J. (2022). Implementing the African Continental Free Trade Area: A Simple, Scalable, and Fast Computational Approach for Algorithmic Governance. In F. Olayele & Y. Samy (Eds.), Sustainable Development in Post-Pandemic Africa: Effective Strategies for Resource Mobilization. Routledge. (Atkinson & Potvin, 2022)
https://www.routledge.com/Sustainable-Development-in-Post-Pandemic-Africa-Effective-Strategies-for/Olayele-Samy/p/book/9781032027609

Abstract: “This chapter suggests a rationale for a simple, scalable, and fast computational approach through the ‘Data With Direction Specification’ (DWDS) to supplement the implementation of the African Continental Free Trade Area (AfCFTA). The specification provides a way for digitally executable versions of rules to be published on the Internet in a platform-agnostic open standard format, across all types of rule-makers and rule-takers, together with the means to allow efficient discovery and transmission of information about rules that are ‘in effect’, ‘applicable’ to any category of transaction, and to be ‘invoked’ by a particular transaction. Relations of obligation, permission, and encouragement can be expressed and understood in natural language, including any vernacular languages, of each stakeholder and equivalently by their heterogeneous computational systems. The resulting ‘Internet of Rules’ (IoR) is intended to enable computer-assisted rules-based coordination for human-centred algorithmic governance. To bolster the resilience of Africa’s markets to social, ecological, and epidemiological disruption, such a ‘Trade Policy 3.0’ approach would make it possible for users to automatically fetch rules via applications and, at the discretion of the parties, invoke rules to digitally automate cross-border compliance in alignment with the policies of national jurisdictions, Regional Economic Communities, and the AfCFTA framework.”

Potential Use of DWDS for Monetary Anchoring

Pringle, R. (2019). The Power of Money: How ideas about money shaped the modern world. Palgrave Macmillan. p. 283. (Pringle, 2019)
https://link.springer.com/book/10.1007/978-3-030-25894-8

My design of the Earth Reserve Assurance (ERA) monetary framework requires a decentralized distributed platform-agnostic system for normative and empirical data distribution and processing. The excerpt below does not mention this deployment consideration, however several entries on the Xalgorithms Foundation blog outline experimental implementation work related to DWDS implementation. e.g. https://xalgorithms.org/writing/watch-the-era-demo-presentation

"An original approach is pioneered by Joseph Potvin in his proposal for an Earth Reserve Assurance (ERA). This is a framework for valuing assets, including currencies, in a multi-currency system with no central reference unit of account. ERA does not itself create a currency unit. It is a new type of primary commodity reserve system. The method of valuation is designed to mirror the long-term capacity of a currency region to produce primary commodities. It uses practical and measurable factors such as topsoil volume, fertility and distribution, fresh water availability, quality and regularity, various ores for metal and minerals, species populations, genomic diversity and integrity, the extent and condition of local, regional and global habitats, essential biogeochemical cycles, and other indicators of sustainable productive capacity. Assurance of this capacity in the form of Earth Reserve deposit receipts—audited by independent certified authorities, and issued by banks—would serve as collateral for a market in these receipts. Each currency obtains its own Earth Reserve Index. As the index accorded to each currency changes, a participating currency becomes more expensive or cheaper depending on whether ecosystem integrity and resource availability are worsening or improving within each currency zone. A currency becomes more expensive as the Earth Reserve is undermined in the areas where it is used. A currency becomes more affordable as ecosystem integrity and resource availability are enhanced in areas where it is used. Potvin argues that this would create a dynamic force in global trade that is the opposite to what occurs presently. Income and jobs will generally migrate towards regions that enhance the Earth Reserve. (Potvin, 2019a)

Potential Use of DWDS for Regulatory Operations Management

WEF. (2022). Regulatory Technology for the 21st Century. White Paper. World Economic Forum, and the Global Futures Council on Agile Governance. (WEF, 2022, p. 12) *https://www3.weforum.org/docs/WEF_Regulatory_Tech_for_the_21st_Century_2022.pdf*
[See Section 7.4.4 *Common Sense-Making Through a Period of "Incommensurable Paradigms".] The excerpt below refers to "Oughtomation", the name I created and used between 30 December 2019 and 15 October 2021 for the system being designed through my doctoral program. I changed the name to the "Data With Direction Specification (DWDS)" in late 2021. Evidently the authors of this report relied on an earlier version of my draft dissertation.]*

"Case study Xalgorithms: Oughtomation [Data With Direction (DWDS)]

Regulations can be difficult to implement in a technical system, but solutions like Xalgorithms provide innovative solutions to interpret and comply with these rules.

A. Problem

A need was identified to automate high-precision rules, such as jurisdictional taxation, efficiently.

B. Action

Who: Xalgorithms

What: Applying dynamic ecological constraints (rules) was identified as an effective way to manage/execute regulatory processes. However, fiscal and regulatory methods were too blunt to be applied at the time. "Oughtomation" [DWDS] is a general-purpose request-response messaging system under free/libre/open licensing for use across any digital network.

Challenges: Balancing the time required to develop interest from diverse individuals and organizations with the need to focus on system development and documentation presented challenges.

Technical Elements:
– IPFS or Cassandra
– Rust
– Electron/react
– Diverse data sources

Impact: Deployment of "Oughtomation" [DWDS] will make it easy for anyone to publish, discover, fetch, scrutinize and prioritize rules, to be directly read and understood by non-specialized humans and machines, for any purpose, in any language. It can also lower the costs associated with interactions across commercial systems.

C. Insights

Multidisciplinary Teams: Cross-division teams help convert rules into a form that can be discovered and leveraged by individuals who are not specialized, as well as machines.

Interoperability: The ability to use the system across different forms of regulations and systems is important, especially when navigating a fragmented regulatory framework.

Stakeholder Buy-In: Receiving buy-in from regulators and governments to support innovative RegTech initiatives is an ongoing challenge.

Open Relationship: Open relationship Ensuring free/libre/open relationships was identified as a key success factor.

Design principles: Following a set of design principles (i.e. simplicity, intuitiveness, decentralization, modularity, least power, tabular declarative style) makes a significant impact on the user experience.

Other: Additional key success factors included directing the team's focus to human-centred automation and being a tolerant brand that is driven by community.

Stalnaker, S., Murray, W., Johnson, J., & Potvin, J. (2016). ***RAIN and RAIL (Real-time Asset Interchange Network on a Real-time Asset Interchange Ledger)*****. Submission to the "Qualified Independent Assessment Team" (QIAT), US Federal Reserve's "Faster Payments Task Force" (FPTF) convened under contract by McKinsey & Company. (Stalnaker et al., 2016, p. 6-7)**
https://fasterpaymentstaskforce.org/wp-content/uploads/hub-culture-vs.pdf

Through 2016 and 2017 I was a member of the Faster Payments Task Force (FPTF) of the US Federal Reserve, through the not-for-profit Xalgorithms Foundation which I co-founded to manage the system design and components arising from my doctoral research. I presented my concept about an Internet of Rules, via a short video (Sandiford, 2016, 2021), as part of the "Solutions Showcase" hosted by the task force. This led to my being invited to integrate my early design concept into one of the formal proposals for an end-to-end payments solution. The final report of the task force explains the context:

> *"In early 2016, the task force solicited proposals for end-to-end faster payments solutions that could address the need for safe, ubiquitous, faster payments. Seeking to address potential conflicts of interest, as well as concerns that all task force participants might not be qualified to assess the proposals, the task force recommended establishing an external Qualified Independent Assessment Team (QIAT) to conduct objective proposal assessments. On behalf of the task force, the Federal Reserve selected McKinsey & Company to conduct a comprehensive assessment of each solution against the task force's Effectiveness Criteria. Rather than ranking proposals or endorsing any particular solution(s), the assessment process was designed to make all solutions better by enabling each of the proposers to iteratively refine and improve their proposals.*
>
> *The QIAT reviewed 22 proposals and 19 proposers opted to continue the process of task force review. For this review task force participants provided solution-enriching feedback on their proposals and the QIAT assessments, as well as overall feedback on the process. After receiving comments from the full task force, 16 solution proposers decided to release their proposals to the general public. These proposals and assessments provided important input for this report's development." (Faster Payments Task Force, 2017, p. 9, 10)*

The following excerpt is from our joint submission posted publicly by the US Federal Reserve:

"Predicability: How does the Solution ensure that all aspects of the payment experience comply with applicable consumer protection requirements, regulations, and commercial laws?

Many rules come into transaction from sources other than the payer, the payee or the intermediary. Rule sets such as duties and taxes, with all their exemption and credits, are very complicated, and are also complex in the sense that they change though time in ways that parties cannot always anticipate.

Laws and consumer protection requirements vary widely by jurisdiction. Fundamentally, the Unique Synchronized Identity (USI)'s strength lies in its flexibility, and the ability to unlock benefits as more data is added, allowing the user or the issuing NAP the capability to adjust services based on these considerations. The possession of a USI in its most basic form may not be enough to enable a financial payment, but it sets the framework for more data to be added to allow a payment in a consistent format when such requirements are met.

Xalgorithms controls and components (under current applied research, design, prototyping and testing) will enable any digital commerce or payment solution to draw reliably upon standardized external computational rules in a common way.

Cross-border functionality: How will the Solution manage conversion of multi-currency funds?

The system envisions the use of multiple currency feeds to determine pricing of multi currency assets relative to the MBA. Conversion between the MBA and the multi-currency funds could be enabled via real-time pricing from these feeds. The system is not designed to hold multicurrency local assets in the pool. However once operational, Xalgorithms rule controls and components may be useful towards auto-generating specialized payment instruments in distributed secure client environments, with full re-calibration traceability."

Potential Use of DWDS for Dynamic Pricing in Smart Contracts

TransportXtra. (2019, February 4). Rail-Powered Property - Property-Powered Rail: A transformational approach? (TransportXtra, 2019) *hps://www.transportxtra.com/publications/local-transport-today/news/60202/rail-powered-property--property-powettred-rail-a-transformational-approach-/*

My design of the 'Rail-Powered Property – Property-Powered Rail' (RPP-PPR) Open Market Development Model" requires a decentralized distributed platform-agnostic system for normative and empirical data distribution and processing. The excerpt below does not mention this deployment consideration, however the associated biographical note refers to the "financing model, the revenue stream for which makes use of the emerging tech 'Internet of Rules' (IoR) system specification and components" arising from my doctoral research at Université du Québec.

"Between 31 May 2018 and 31 July 2018, the Department for Transport held a 'Call for ideas' in respect of rail market-led proposals (MLPs); essentially a call for proposals that were 'financially credible without government support'. ...

In Canada's National Capital Region (over a dozen municipalities including Ottawa Ontario and Outaouais Quebec), a consortium of companies is taking a market-led design from a business design innovator to its logical conclusion. The 'Rail-Powered Property – Property-Powered Rail' (RPP-PPR) Open Market Development Model is designed to optimize return on investment in property by delivering metropolitan-scale passenger railway systems and services on a commercial open market basis, without dependence upon government subsidies, public debt or taxes.

Joseph Potvin will be speaking and running a Q+A session at the Rail Stations and Property Summit on February 27 in London. ... A key enabler of the process is the sharing of value-added in transactions through a 'smart-contracts' component co-designed by Potvin in his capacity as Executive Director of the Xalgorithms Foundation, a Canadian high-tech not-for-profit that is dedicated to 'the development and evolution of a free/libre/open, standard way to publish and to fetch computational algorithms over the Internet, with particular attention to advancing the fairness and efficiency of markets through algorithmic commerce'.

Essentially, a passenger railway project management company plans a high quality train service, then in an open market it sells licenses to independent station enterprises. Under a framework agreement, trains service each participating locality in exchange for a monthly fee based upon a formula that makes use of independent empirical data on the property market effects of that railway access.

The increment attributed to the railway service by a mutually agreed property valuation company sets the price, but each station enterprise remains free to assemble funds by their preferred method (for example, affirmative covenants; common elements freehold condominium; municipal tax instruments).

This RPP-PPR model can bootstrap all start-up funds for the railway and generate sustained income, at a scale enabling straightforward capital financing of infrastructure. This eliminates dependence on government subsidies, public debt or taxes, and it directs investors towards diverse property development opportunities around stations.

This model is designed to respect each municipality's prerogatives over zoning, approvals, taxation, bylaws and municipal transportation planning in accordance with normal application review and approval processes, whilst providing for rail systems that would normally involve massive public sector spending, and tricky co-operation among several municipalities."

Speaker: Joseph Potvin leads design of the "Rail-Powered Property ⇄ Property-Powered Rail" (RPP⇄PPR) financing model, the revenue stream for which makes use of the emerging tech "Internet of Rules" (IoR) system specification and components which he co-designed. ... He holds an MPhil from Cambridge (geographical economics) and HonBA (economics) from McGill. His current doctoral research at UQuébec involves meso-level systems design for enhancing micro-level performance that can proliferate to achieve desired transformative macro-level emergent effects. *http://landor.co.uk/railstations/2019/speakers.php*

Kelter, J., Conboy, W., Wit, J., Wilensky, U. & Potvin, J. (2022). A General-Purpose 'Economic Petri Dish' ABM With Land, Labor, Capital, and Organization to Test Four Ways to Index Prices to Economic Fundamentals (Poster Submission). Computational Social Science Society of the Americas (CSS2022) Annual Conference, Online. (Kelter, Conboy, et al., 2022)
https://computationalsocialscience.org/conferences/css2022
Referenced here: http://ccl.northwestern.edu/papers.shtml

This poster provided an overview of our team model. A particular part of this model was also submitted as a full conference paper to the Annual Conference of the Computational Social Science Society of the Americas, and was selected for the "CSS 2022 Best Paper Award", to be published by Springer in 2023:

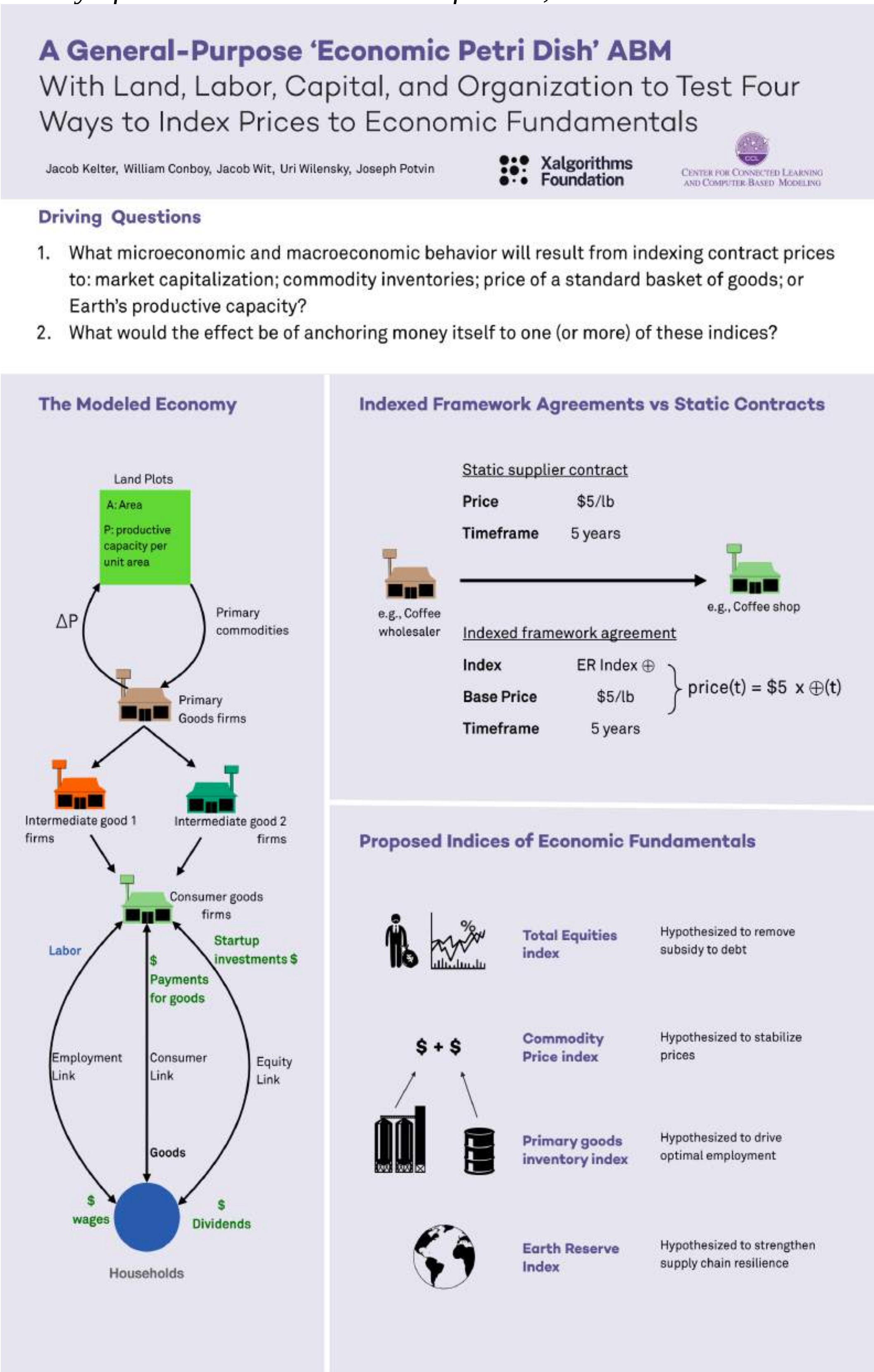
Kelter, J., Wilensky, U., & Potvin, J. (2022). Introducing Land Constraints to Macroeconomic Agent-based Models. Proceedings of the 2022 Conference of The Computational Social Science Society of the Americas. (Kelter, Wilenski & Potvin, 2022)

My role as co-author in both the poster and paper was twofold: (a) principal designer of the system which Jacob Kelter and two interns modelled in Netlogo; (b) substantive refinement of explanations.

This agent-based modeling work is summarized in Section 5.2 and in Table 11 of this dissertation. A later version of this model will be a full DWDS reference implementation. The excerpt below, from the poster submission, does not mention this future deployment consideration. However several entries on the Xalgorithms Foundation blog describe experimental implementation work that is related to DWDS implementation (e.g. *https://xalgorithms.org/writing/watch-the-era-demo-presentation*).

> "Our model is designed to test a choice of indexing methods for supply contracts as described in concurrent research by Potvin [5]. The model still contains a single type of firm selling goods directly to consumers but allows for an arbitrarily complex network of primary and intermediate firms supplying goods to one another, and ultimately, to the consumer goods firms. The purpose of this added complexity compared to prior MABMs (Macroeconomic Agent-Based Models) is to address two interrelated research questions: 1. How does the network structure of firms in the economy affect volatility of macro-level variables? 2. What effect can indexed supply contracts have on both individual firm performance and volatility of macro-level variables? ...
>
> In addition to their relationships with households, firms have trading relationships with one another known as framework agreements, following the structure described by Potvin [5]. A framework agreement specifies an index (e.g., an index of primary goods prices), an index- multiplier and an expiration date. The price at any given time equals the current value of the index times the index-multiplier. If a framework agreement is not indexed, then the price is a fixed dollar amount."

Figure 52: Poster Submission to the 'Computational Social Science Society of the Americas' Annual Conference, 2022



Appendix D: Informal Comments by Technical Contributors

Each incremental draft of this work during the past three years has been posted online under the Creative Commons Attribution 4.0 International License.[86] Through various paths, a small number of individuals have helped to create the first reference implementation of the Data With Direction Specification (DWDS) under the Apache 2.0 License (RuleMaker application, RuleTaker component) and the Affero 3.0 License (RuleReserve network service software). Their implementation efforts along the way have influenced my trajectory as the designer of this system,

In this Appendix I allocate space to three of the principal technical contributors to the first reference implementation. Participation in such free/libre/open informal teamwork involves individuals in their personal capacities, driven by their own interests and views, independently of the organizations in which they currently work or have worked. Their professional backgrounds are mentioned below only to acknowledge their subject matter competence, and for transparency. This appendix is limited to individuals who are 'arms length' from financial involvement in the research, other than as explained in associated footnotes.

Via email in July 2022 I asked these three contributors to "please write a short testimonial about your technical view of what I have designed with team input". Specifically I invited them to comment on aspects of the specification which relate to their respective implementation roles. Each of their replies was first posted online in the team workspace hosted by GitLab (Cunneyworth, Kelly & Kim, 2022) The body of the present version of the dissertation has not been updated to incorporate any of the suggestions or ideas arising from their comments appearing below, in order to avoid circularity among comments and adaptations at this point.

- **Wayne Cunneyworth** is a veteran data scientist, retired from DataKinetics. The tabular declarative data processing system for mainframes that he co-designed with William Olders[87] provides a high-performance rules engine used by many of the world's largest banking, credit card, brokerage, insurance, healthcare, retail and telecommunication companies. Though his manual "Table Driven Design" and through conversation, Wayne helped me to understand tabular declarative data structuring and processing, and he suggested I consider four-element logic for DWDS.

86 Within the methodology portion of the present dissertation, Section 2.5.1.2 explains the rationale for "Free/Libre/Open Relationships" as a "design virtue".

87 William Olders, is President of DataKinetics, which has contributed financially to this design research. He and the author of this dissertation are co-founders of Xalgorithms Foundation, which was incorporated to provide the organization base for the reference implementations of the system design described herein.

- **Don Kelly** is a full-stack informatics designer/developer, currently working as the coordinator of the "developer environments" team at Shopify, which is responsible for the day-to-day working environments used by all developers at Shopify. That undertaking is summarized at https://shopify.engineering/shopifys-cloud-development-journey. Don has provided team guidance for DWDS reference implementation versions 1.x through the present 3.x,[88] and he is the lead implementer for RuleReserve network service and the RuleTaker component.
- **Ted Kim** is a senior data scientist and full-stack informatics designer/developer in the regulatory section of Health Canada, for pre-market prescription drugs. Ted initiated the implementation of RuleMaker version 3.x. as a Web app with a Javascript framework, and he has guided technical aspects of the ongoing RuleMaker development work in the Svelte environment by undergraduate student Huda Hussein. Ted has also assisted with 'data wrangling' experiments related to the Earth Reserve Assurance (ERA) use case to be built upon a DWDS infrastructure.

Wayne Cunneyworth, Hons BSc MSc (Computer Science)
Cherryville, BC, Canada
wayne@cunneyworth.com
https://gitlab.com/xalgorithms-alliance/data-with-direction-specification/dwds-documents/-/issues/3

Some Thoughts on Multi-Valued Logic

"In June 2020 I suggested Joseph consider four-valued logic for his tabular logic gate. In particular I sent him a link to the article "*Beyond True and False*" by mathematician/philosopher Graham Priest, who is currently a professor at Ruhr University of Bochum, Germany. (Priest, 2014) So I will focus my comments upon this theme.

Ever since my favourite courses in Switching Theory, Computer Logic Design, Computability Theory and Artificial Intelligence at the University of Manitoba back in the early 1970s, through subsequent years as a rules automation software consultant for several global Fortune 500 companies and governments, I have had a special interest in innovative software designs that combine rule-based structures with attributes of high performance and ease of maintenance.

In reviewing sections of Joseph's "Data With Direction" thesis of 22 June 2022, I have the following comments:

First, from a functional perspective, a tabular declarative implementation of rules is clearly equivalent to a procedural implementation. Decision tables have been around for a long time and may seem antiquated to modern programmers with expertise in the most popular modern programming languages. But procedural and tabular designs are complementary, not incompatible, and tabular rules continue to be effective in today's systems. The most obvious advantage to using external tables for rule details, independent of the generalized procedural code, is ease of maintenance for the separate rules-based component, with little or no data coupling impact on the generalized procedural component.

88 Don Kelly worked under paid contract to Xalgorithms Foundation in 2016 and in 2018, throughout the early conceptualization of "an Internet of Rules".

Second, multi-valued logic, including quaternary systems, is an established field of study and a rich area for new research. There can be various useful interpretations of values beyond True and False that are also mathematically coherent. I suggest Joseph consider submitting a paper to the *International Symposium on Multiple-Valued Logic* planned for May 22-24, 2023.[89] This conference series has been convened annually since 1971 by the Technical Community on Multiple-Valued Logic of the IEEE Computer Society. He could also submit an article to the Journal of Multiple-Valued Logic and Soft Computing.[90] A useful review of this field is provided by Giovanni Panti (Panti, 1998):

> Panti, G. (1998). Multi-Valued Logics. In P. Smets (Ed.), Quantified Representation of Uncertainty and Imprecision (pp. 25–74). Springer Netherlands. *https://doi.org/10.1007/978-94-017-1735-9_2*

In section 4.2 of Joseph's dissertation, the section entitled *'Available Methods for Logic Data Models'* provides ... an important distinction between {T, F, N} and {T, F, -}. I find that Joseph's explanation for the "N" element invites some confusion, particularly in his comment: "not knowing whether a value is T or F is entirely different than knowing that a value is not T or F". While this comment is not wrong, I think it could be tightened up for clarity. I think of the distinction as follows:

- *"N" (don't know) implies "not knowing whether a value is T or F or even if it is relevant for invoking the rule"*

This is entirely different than:

- *"-" (don't care) implies "knowing whether a value is T or F but either one is acceptable and both are known to be relevant for invoking the rule".*

The footnote 18 in Joseph's thesis, where he suggests interpreting N to flag a quality issue is a good point.

Happily, the aforementioned concept of relevance is addressed very effectively in section 5.3.9.3 of his thesis, with the title: *"Discussion: The 'DWDS Logic Gate' Differs from a 'Decision Table'"*. [H]e writes: "To be uncertain about everything represents a lesser state of knowledge than to be uncertain about something definite". Furthermore, the utility of his particular quaternary logic design is clearly oriented to server the purpose of a DWDS logic gate, namely,

- *"to inform [agents] about [possibly multiple] rules which are [in effect] in such a way as they are more informed while making their decision" as opposed to the purpose of a Decision Table - or any equivalent procedural specification - with "only one outcome".*

Regarding Joseph's use of an element for a simultaneous "Yes AND No", initially it was not clear to me why he required this. But his example ... of the California law which states that a bumblebee is a fish provides a straightforward way to see the utility of this element in applied circumstances. It is common in real law, policy and business that contradictory states occur and persist, even after the courts 'resolve' the conflicts, as in the California illustration. I was interested in Joseph's pursuit of this into Bertrand Russell's work and others.

Generally I think the handling of uncertainty with multi-valued logic is a very important area of applied computing that merits greater research and development effort. It seems to me that Joseph makes a credible, useful and very accessible contribution that will attract further interest among formal logicians.

I think Joseph makes a credible case for using a tabular design with quaternary logic in a flexible advisory role for human or machine agents.

Wayne Cunneyworth

89 http://mvl.jpn.org/MVL/event.php

90 http://www.oldcitypublishing.com/journals/mvlsc-home/

Don Kelly, Hons BSc (Mathematics)
Ottawa, ON, Canada
karfai@gmail.com
https://gitlab.com/xalgorithms-alliance/data-with-direction-specification/dwds-documents/-/issues/2

Programming a Simple Transferable Atomic Data Package for 'Data Logic and 'Business Logic'
RE: "Data with Direction" Thesis (June 2022) by Joseph Potvin, doctoral candidate, UQO

I've been collaborating with Joseph Potvin off and on for a number of years, exploring how to implement his emerging design. Now that he has completed his thesis I'd like to offer my own take on the substance of his 'Data With Direction' concept, grounded in my own purposes and perspective as a systems implementer.

I'll frame my comments in the form of an initial draft of short paper, intended as a contribution towards attracting a wider community of systems implementers. I've tentatively entitled this paper: "Programming a Simple Atomic Data Package Using DWD for Simple Transferable 'Data Logic' and 'Business Logic'". Essentially I present a direct technical illustration of the utility of the "Data with Direction" design described and explained in the June 2022 version of Potvin's doctoral thesis at Université du Quebec en Outaouais (UQO), supervised by Dr. Stéphane Gagnon.

I've worked for 25 years designing and building software solutions and leading teams. My latest endeavour has been assembling a "developer environments" team at Shopify. This group of less than a dozen full-stack developers builds foundational software used by all developers at Shopify for their day-to-day working environments. You can read about my most recent work at https://shopify.engineering/shopifys-cloud-development-journey. Previous to this, I led project teams at an outsourcing company. In that role, I'd regularly bootstrap new projects for a wide variety of clients. I've helped build mobile phones, e-readers, smart televisions, medical case management applications, e-health platforms, and data analysis frameworks.

Don Kelly

Programming a Simple Atomic Data Package Using DWD for Simple Transferable 'Data Logic' and 'Business Logic'

A Technical Illustration of the Utility of the"Data with Direction" Design Described in a Thesis by Joseph Potvin, Doctoral Candidate, UQO

Don Kelly (karfai@gmail.com)

June 26, 2022

Joseph Potvin's design for rules expression[1] solves a problem that I've encountered many, many times in my career. The "Data With Direction" (DWD) concepts pull what the industry calls "business logic" out of a mire of particular implementations into a cleaner, transferable specification that can easily be integrated into other solutions.

Why is this important? Many of the solutions we implement intersect with the "real world" where "real rules" come into play. From the e-commerce domain, where I'm currently focused, this might include rules related to domains like taxation, payments, lending, or logistics. When we implement these solutions, we embed interpretations of these rules into the software that we write. These interpretations are completely disconnected from the interpretations of other implementations as well as their "real world" counter-parts. A solution that incorrectly interprets taxation does not forgive the user their obligation to a "real world" tax authority. Different implementations that behave differently erode user trust in all implementations.

This is a bold, idealistic statement but how would it work in a real implementation? Let's consider a payroll solution implementation for a company operating in Canada. We'll look at a common aspect of such payroll solutions: tax withholding. This solution encodes "work-related" rules about part-time, full-time, and contract workers.

[1] In his thesis Potvin acknowledges direct contributions to this particular method from Bill Olders and Wayne Cunneyworth as well as the wider literature in tabular declarative programming, most prominently work by David Parnas, Jan Madey and Michal Iglewski in the 1990s, and Art Lew in the 1980s.

This will be a simplified example used to illustrate the concepts of DWD and shouldn't be considered to represent a "complete" solution.

The current practice for building such a solution would retain persistent data in a Relational Database (RDBMS) and use forms of "domain modeling" to represent data from the RDBMS using classes in an object-oriented programming language (OOPL). Given this understanding, we can assume that such a solution would include classes for our three types of employees. Let's call them `PartTime`, `FullTime`, and `Contract`. In typical fashion, these classes might be derived from a generic base class. Let's call that `Employee`.

To simplify this example, let's assume there are two levels of taxation with a single tax percentage for each level of daily compensation, rather than accumulating percentages:

- below 500$, there is no tax withheld
- above 500$, but below 2000$, 10% of the gross pay is withheld
- above 2000$, 25% of the gross pay is withheld

We might begin to implement the concept of withholding in the `Employee` base class. This seems like a good decision because the calculation of "net pay" is applicable to all of the employees in our model.

Given these considerations, we might implement the functionality as:

```
class Employee
  def net_pay()
    return gross_pay() - withheld_tax()
  end

  def withheld_tax()
    if gross_pay() < 500
      return 0
    else if gross_pay() < 2000
      return gross_pay() * .1
    else
      return gross_pay() * .25
    end
  end
end

class FullTime < Employee
  def gross_pay
    # calculated based on daily salary
  end
end

class PartTime < Employee
  def gross_pay
```

```
    # calculated based on hours worked
  end
end
```

This implementation doesn't take `Contract` class into consideration. Those workers pay their own tax, so nothing should be withheld. This could be implemented by overriding the calculation in the derived class:

```
class Contract < Employee
  def withheld_tax()
    return 0
  end
end
```

An experienced coder, during code review, would have some immediate concerns with the long term maintainability of this code, as written. While this code's adherence to "the rules" is likely correct, for now, the implementation, as written, contains a few immediate values that a reasonable coder will expect to change. Since changing these values would require the implementation to be re-built (or re-deployed), the immediate review comment would recommend that these immediate values be moved to application data, rather than application code. Since the current fashion of application development typically retains data in an RDBMS, the impetus would be to extract the immediate values to a DB table. Likewise, a more experienced coder would expect that even the classifications of employees had the potential to also require change over time.

A common technique for extracting logic from classes into database tables would be an object-relational mapping (ORM). Under this technique, the various classes in our object-oriented design would be associated with unique tables. A common base class (`Employee` in this example) would itself have its own table. The ORM framework in use would perform the necessary association between the table of the derived class and that of the base class. Note that the classes in this example are used as "structural conditionals". This means that the calculation could be implemented procedurally. Therefore, in order to skip directly to the point, we'll express the calculation in that manner to avoid the unwinding of what would be a quickly cumbersome collection of data tables for such a simple example. We might express `withheld_tax` procedurally as:

```
def withheld_tax(employee_type, gross_pay)
  if employee_type == :contract
    return 0
  else
    if gross_pay < 500
      return 0
    else if gross_pay < 2000
      return gross_pay * .1
    else
      return gross_pay * .25
```

```
    end
  end
end
```

From this procedural implementation, we can easily derive a database table[2] to represent the "data logic":

type	threshold	percent
contract	999999	0
fulltime	500	0
fulltime	2000	0.1
fulltime	999999	0.25
parttime	500	0
parttime	2000	0.1
parttime	999999	0.25

This allows a simplification of our original procedure (this pseudocode assumes an RDBMS library):

```
def withheld_tax(employee_type, gross_pay)
  vals = db.execute(
    "SELECT percent FROM withholding
       WHERE #{gross_pay} < threshold AND #{employee_type}=employee_type"
  ).first
  return gross_pay * vals.percent
end
```

This refactoring of the original solution satisfies the typical requirements of the trade today. However there's still a problem with this implementation. While we've successfully extracted the "data logic" from the code, the actual "business logic" remains. The table, as expressed, gives some hints to how the data should be interpreted, but the table is not the final authority. That still lies in our implementation. If an authority of the "business logic" were to require a change, we'd be back to reimplementing. Ideally, if that authority required a change, *they* would be able to provide it, requiring *minimal* reimplementation on our part.

Potvin's DWD design gives us the final step in this refactoring process. As seen above, most of these refactorings would halt once we'd achieved a "data logic" implementation that could react to most necessary changes. As stated, this still lacks the capability of transferability. Potvin's design unlocks that capability by also encoding the "business logic". Following is a translation of

[2] This example table uses a maximum value sentinel (`999999`) to simplify the SQL query. There are better methods for designing the table to avoid this sentinel value but employing them here would obfuscate the point being made.

this implementation into DWD[3]:

```
SCENARIOS
                                [A,  B,  C,  D,  E,  F,  G ]
CONDITIONS
  employee_type='contract'      [01, 00, 00, 00, 00, 00, 00]
  employee_type='parttime'      [00, 01, 01, 01, 00, 00, 00]
  employee_type='fulltime'      [00, 00, 00, 00, 01, 01, 01]
  gross_pay<500                 [00, 01, 00, 00, 01, 00, 00]
  gross_pay<2000                [00, 00, 01, 00, 00, 01, 00]
  gross_pay=>2000               [00, 00, 00, 01, 00, 00, 01]
```

dissertation steps beyond fully-determined logic, to accomodate various types of uncertainty. He provides a simple unified method to handle both deterministic and complex programming requirements.

As an aside, you'll notice that this implementation of a "rule" in DWD includes arithmatical expressions in *both* the `CONDITIONS` *and* `ASSERTIONS` (e.g. `withheld=gross*.25` or `gross_pay=>2000`). This facility is not typically shown in the examples used in the dissertation. This might have been a deliberate decision on Potvin's part to emphasize the particular sentence formulation described in the thesis. However I'd recommend that some examples of this level of elaboration be added so that the reader might understand the full capabilities of the DWD expression format.

Either way, it should be evident from the factoring progression shown in this response that Potvin's DWD concepts are *fully capable* of expressing the "data logic" and "business logic" typical of applications implemented in industry today.

Ted Kim, Hons BSc (Genes, Genetics and Biotechnology)
Ottawa, ON, Canada
ted.kim03@gmail.com
https://gitlab.com/xalgorithms-alliance/data-with-direction-specification/dwds-documents/-/issues/1

Sentiment and Intent Behind DWDS RuleMaker Reference Implementation
RE: "Data with Direction" Thesis (June 2022) by Joseph Potvin, doctoral candidate, UQO

Through my years working at Health Canada, both as a regulator of pre-market prescription drugs and now as a digital solutionist, I have become increasingly concerned that our rules (legal frameworks and regulations) may not ever be nimble enough to adapt to the pace of innovation.

Therefore I personally welcome the experimentation with DWDS. Joseph's contributions have paved the way for many rule systems across the globe to work coherently and in a common way across diverse platforms. Particularly as a regulator, I consider that the system he has designed provides a viable opportunity for fully expressing our complex set of regulations in a "processable" manner where its applicability can be digitally computed.

Perhaps the most noteworthy factor of all from a regulator's perspective would be that DWDS provides a pathway for vastly minimizing the administrative burden required for compliance and enforcement. Permit me to illustrate this with a genuine example of the complicated requirements faced by both the regulators and regulated in the pharmaceutical industry. The table below is from Health Canada’s Post-Notice of Compliance (NOC) Changes: Quality Document. (Health Canada, 2019). This table identifies what is expected each and every time there is a change in the specification for any drug substance that requires structured testing and acceptance criteria. The document also states: “Guidance documents are administrative instruments not having force of law and, as such, allow for flexibility in approach. Alternate approaches to the principles and practices described in this document may be acceptable provided they are supported by adequate justification.”

Considering that this table is just one of 245 similarly structured tables in a massive reference document, it is entirely impractical to write a conventional software program to automate the requirements validation, especially considering that each affected logic must be reprogrammed as the document evolves.

However it appears that Joseph has shown a way to create a workable ‘rules-as-data’ network approach to the problem. I have been contributing to the implementation the DWDS RuleMaker application because I see how the structure and method he has designed can enable two or more entities to express and agree upon a set of rules to apply, in a way that the outcome can be quickly reviewed by a third party. It is not forceful and leaves room for ambiguities to be resolved, which is common in a policy context, and hence, why I agree that incorporating uncertainty into his logic gate is important for rule expression. With his design a rule can be much more simply reviewed, revised and its outcome implications can be instantly "previewed". Comparisons among versions and alternatives are very easily accomplished. Where compliance fails, the issue will only implicate the specific rows or columns of the logic gate without jeopardizing other systems. It becomes much easier for all the involved parties to maintain an integrated set of rules across organizations and jurisdiction. Moreover, DWDS also provides the path forward for international organizations to apply consistent sets of rules that span many lexical diversities across jurisdictions, without forcing any rule to be expressed in a fixed standard language.

Excerpt: "Change in the specification for the drug substance involving test and acceptance criteria", in: Post-Notice of Compliance (NOC) Changes: Quality Document Appendix 1, Section 3.2.S, Table 8, pgs 36-37

Description of Change	Conditions to be Fulfilled	Supporting Data	Reporting Category
8. Change in the specification for the drug substance involving test and acceptance criteria:			
a. for sterile drug substances, replacing the sterility test with alternate microbiological methods or process parametric release	None	1-7	Supplement
b. deletion of a test	1-5	2, 7-8	Annual Notification
c. replacement of a test	1-6	2-5, 7-8	Annual Notification
d. addition of a test	None	2-5, 7-8	Annual Notification
e. relaxation of an acceptance criterion	1-4, 6	2, 7-8	Annual Notification
f. tightening of an acceptance criterion	None	2, 7-8	Annual Notification

Conditions

1. The change is not necessitated by unexpected events, resulting in failure to meet specifications, arising during manufacture or because of stability concerns.
2. No change in the polymorphic form and impurity profile that impacts safety or efficacy of the drug product.
3. The change does not concern sterility testing.
4. The change concerns drug substances that are discrete chemical entities (i.e., this does not include polymeric complexes).
5. The deleted test has been demonstrated to be redundant with respect to the remaining tests and does not impact the safety or overall quality of the product (e.g. removal of an organic volatile solvent test after at least 10 commercial scale batches tested and meet acceptance criteria, or provide valid scientific justification).
6. The relaxed criterion is in accordance with compendial and/or ICH criterion.

Supporting Data

1. (S.2.5) QC approved process validation and/or evaluation studies or the proposed validation protocol of the proposed drug substance.
2. (S.4.1) Updated, QC approved, proposed drug substance specification.
3. (S.4.2) Copies or summaries of analytical procedures, if new analytical procedures are used.
4. (S.4.3) Copies or summaries of validation reports, if new analytical procedures are used.
5. (S.4.3) Where a House analytical procedure is used and a Schedule B standard is claimed, results of an equivalency study between the House and compendial methods.
6. (S.4.4) Description of the batches, certificates of analyses for one batch, or batch analysis report and summary of results, of a sufficient number of batches (minimum of ten batches) to support the process parametric release.
7. (S.4.5) Justification of the proposed drug substance specification (e.g., test parameters, acceptance criteria, or analytical procedures).
8. (P.2) Where appropriate (e.g., for a change in particle size limit for a poorly soluble drug substance), comparative, multi-point dissolution profiles in the release medium for one batch of the drug product using material from the approved and change drug substance specifications.

This example is a relatively simple one. In fact, Health Canada's regulators and pharmaceutical industries must work with a complexity up to two additional orders of magnitude on a daily basis. Precious time is spent not *assessing the science and data* but instead, merely *assessing administratively which conditions and supporting data are required under a given circumstance*.

Therefore it seems to me that DWDS comes at an opportune time when nations across the world are in the midst of transforming their respective governments to digitized modes of work. Recently several countries signed a Memorandum of Understanding “that they will each promote good practice on rulemaking within their jurisdictions that supports responsible innovation and entrepreneurship while serving citizens’ interests”. The participants agree that: "A more agile approach to rulemaking is needed in order to unlock the potential of innovation and shape it in a way that protects citizens and reflects their values" and that "International co-operation is important to share knowledge and evidence and avoid unnecessary divergence in rules that inhibits cross-border innovation and hinders joint action to address common risks." (TBS, 2020).

Ted Kim

Appendix E: Recent Literature on Project Management Informatics

Afterword, December 2022

Informatics in Recent Academic Journals of General Management and Project Management

Between 2016 and 2022 the design research detailed in the present dissertation produced a rationale and specification for a decentralized distributed platform-agnostic system for normative data distribution and processing, which I named ‘an Internet of Rules’. This emerged simultaneously with a new focus on informatics in the wider academic literature on project management, referred to by terms such as “digitization”, “digitalization”, and “digital transformation”. Figure 50 is adapted from a graph published in a 2022 article by Andrew Temnikov and Amy Podshivalova, who performed a "quantitative analysis of scientific papers devoted to digital transformation in the field of economics, management and business”, by querying three journal databases (Web of Science, Scopus and VAK) with the key phrase "*digital transformation*". Prior to 2016 the total number of articles using this phrase was, in their words, “insignificant and irregular”. But from 2019 forward there have appeared about 1,500 such articles annually on this theme, even amidst the general decline among all types of academic publication activity in 2021 due to widespread disruptions across all sectors. (Temnikov & Podshivalova, 2022, p. 125)

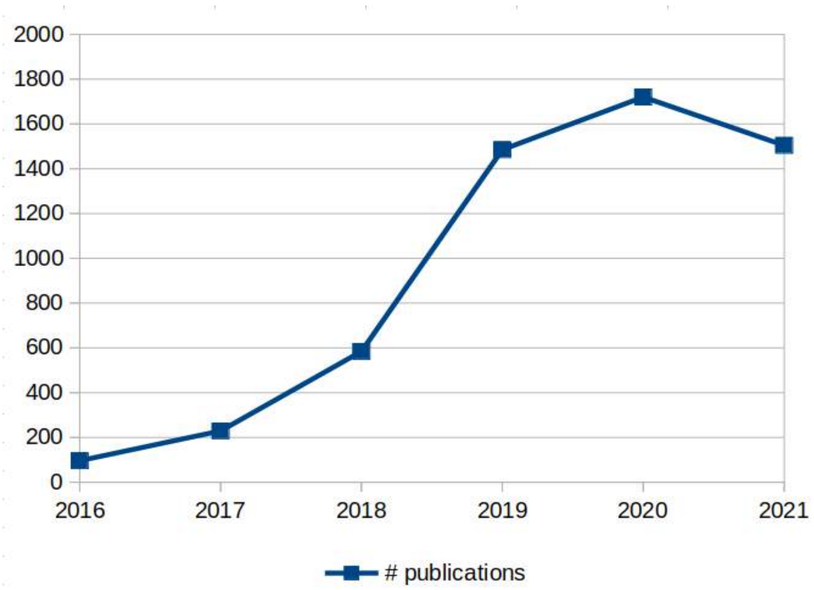


Figure 53: Number of articles found with the phrase "digital transformation" in three academic journal databases: Web of Science, Scopus and VAK. Adapted from Temnikov and Podshivalova, 2022.

The ‘objective’ of the design research leading to my own present dissertation is given in the first paragraph of Section 1.1, based on the rationale expressed in my March 2019 Thesis Proposal:

> “This research fills a gap in project management theory and practice which concerns how a project stakeholder is presumed to discover and obtain factual knowledge of the significant rules that are ‘in effect’ for dates/times and prerogatives relating to identities and jurisdictions of a given context; that are ‘applicable’ to the class of endeavour and task being undertaken; and that are ‘invoked’ by a particular circumstance of the moment. Practical people, as individuals and on behalf of entities, need to obtain timely and comprehensible awareness of relevant rules in order to exercise judgment towards actively aligning with them, or to deciding not to.”

Accordingly in my work a method is described to improve the communication of rules as a distinct class of data with direction that can be instantaneously discovered and transmitted over the Internet.

In a 2021 review of management literature, Peter Verhoef et.al. observed that the vast majority of academic articles to that point which focused attention on digital transformation have tended to be context- or case-specific, and as yet, there had not been much published to provide a general characterization of the theme itself:

> "Despite the ubiquity and visible impact of digital transformation and resultant new digital business models, the academic literature has so far paid surprisingly little attention to these developments, only recently starting to address the topics of digitization, digitalization, and digital transformation. Until now, digital change has received most attention within specific business disciplines. ...To the best of our knowledge, there has been no multidisciplinary discussion on digital transformation" (Verhoef et al., 2021, p. 889).

Verhoef et.al. explain the nuanced jargon that arose recently in reference to phases of engagement, namely ‘*digitization*’, ‘d*igitalization*’, and ‘*digital transformation*’ (Verhoef et al., 2021, p. 891):

- ***Digitization:*** Encoding of analog information into a digital format (i.e., into zeros and ones) for computer storage, processing, and transmission; Goal: cost efficiency and cost effectiveness.
- ***Digitalization:*** Implementing methods based on information technology and data structuring to alter existing business processes; Goal: Business process re-engineering.
- ***Digital transformation:*** Pursuing organization-wide and multi-entity change involving development of new business models; Goal: Strategic reframing of the business.

In a 2020 article about learning from digitalization projects, Bassam Hussein et.al. explained that:

> “Although research on digitalization is expanding, it is mostly focused on strategic and transformational consequences of digitalization rather than providing a framework for managing digitalization projects.” (Hussein et al., 2020)

Similarly, Zeljko Tekic and Dmitry Koroteev have commented:

> “Recently booming academic interest in digital transformation aims to provide continuous support to managers in dealing with this important issue. However, as with all new fields, the scholarly literature is characterized by increased variability and diversity ... resulting in an unclear and blurry understanding of the whole of digital transformation.” (Tekic & Koroteev, 2019)

A year earlier, Jennifer Whyte observed that in the academic literature on digital transformation to that point “there is insufficient attention given to the important question of how increasingly pervasive digital information transforms project delivery models." (Whyte, 2019) The same gap was observed by Dimitri Nemirovski in a comparative study in 2021 entitled *Utilization of Elements of Digital Transformation in Project Management* [PM]. He explains that “Digital transformation [DT] is on everyone's lips and increasingly gaining speed” yet “despite this importance, the possible impact of DT on PM success has not yet been elucidated in the academic literature” (Nemirovski, 2021, p. 18-19).

Informatics in Project Management Standards: More Attention but Continued Chagrin

Unlike the academic journals, trade publications have provided empirical evidence of the relation between the digital transformation wave and project management performance. The Project Management Institute’s *"Pulse of the Profession 2021"* annual report described on an online survey of 3,950 project professionals conducted in late 2020 in which managers were asked: “How would you describe the change in your business over the past 12 months compared to the 12 months prior in the following areas?” The leading ‘big change’ selected by 68% of respondents was "digital transformation" (PMI, 2021, p. 3). Also the PMI’s *“Global Megatrends 2022”* report observed that “six megatrends stand out based on their impact and the implications for projects across the world”, and the dominant ‘megatrend’ named was “Digital Disruption”. However many authors on this theme (for example (Magnusson et al., 2022) (Tekic & Koroteev, 2019)) frame the theme with what appears the present author a sort of “automation bias” (Goddard et al., 2012) (Mosier et al., 1996).

Yet by very many empirical accounts, the outcomes of this digital transformation are astonishingly poor. Writing for Forbes in 2022 Corrie Block summarized the risk facing real investors:

> “In 2016, Forbes assessed the risk of failure in digital transformation to be 84%. According to McKinsey, BCG, KPMG and Bain & Company, the risk of failure falls somewhere between 70% and 95%. Clearly, we’re doing something wrong in digital transformation, yet we’re still willing to go through the motions knowing that it’s largely a waste of resources.” (Block, 2022)

There is also an ongoing literature that details the risks that workers face from having their jobs automated, with titles such as “*The future of employment: How susceptible are jobs to computerisation?*” (Frey & Osborne, 2017) and “*The risk of automation for jobs in OECD countries*”(Arntz et al., 2016). (See also (Nedelkoska & Quintini, 2018).)

Considering the evident risks facing both investors and workers in light of this so-called digital transformation, it would be prudent to pursue such projects with caution. In Section 7.4.1 of the dissertation I tabled the following question for future research: “Must we make a blanket assumption that all projects move us ‘forward’? Might one distinguish a ‘forward’ project versus a ‘backward’ project?” The essential normative directionality of projects is emphasized in an article by Jennifer Whyte and Lara Mottee published in 2022, *Projects as Interventions:*

> “While the recent Project Management Body of Knowledge (PMBOK) guide gives reference to value, we feel the project scholarship community needs to more broadly revisit how it conceptualizes a ‘project’. ... We feel scholarship should encompass arguments against projects as well as for projects, and that there is the potential for significant discussion about **what types of projects are useful.”** (Whyte & Mottee, 2022, p. 939, emphasis added)

The present DWDS design research situates ‘human-centred automation’ as a core design principle (Section 2.5.1.1). More generally with a similar perspective, the challenge is expressed by Päivi Parviainen et. al. as “not about turning existing processes into digital versions, but rethinking current operations from new perspectives enabled by digital technology." (Parviainen et al., 2017) Elena Kalyazina explains that this can involve “restructuring of the organization as a whole: its working principles and its structure in accordance with the new information realities”. (Kalyazina, 2021, p. 4752) Kalyazina explains:

> “The digital transformation of companies is carried out in stages, from the automation of business processes to the complete transformation of the managed subsystem of enterprises. ... In project management, there is also a redistribution of competencies. This is more true of the project manager and work package leaders. Knowledge of the subject area of the project, so-called hard skills, digital technologies (artificial intelligence, robotisation of business processes) are taking over. Control and monitoring competences are also replaced by new digital infrastructure tools, specialised frameworks, and embedded tasks in software. They save time in tracking timetables, transferring information to responsible parties, forecasting risks and much more. In this way, the main roles of the project manager, such as coordinator, moderator, leader, are brought to the fore.” (Kalyazina, 2021, p. 4757)

Kalyazina points to the new standard *ISO 21502:2020 - Project, programme and portfolio management - Guidance on project management* which replaces the original 2012 version; as well as the 2021 updated *Guide to the Project Management Body of Knowledge (PMBOK),* the American National Standards Institute (ANSI) standard produced by the Project Management Institute. Both of these references have shifted from project management processes to various project management principles and practices that are required for adaptive flexible team competencies, and resilient post-project outcomes. (Kalyazina, 2021, p. 4762)

It is my assessment that, especially in regard to the co-called ‘digital transformation’, the ISO and PMBOK standards on project management have finally incorporated many of the factors that International Centre for Complex Project Management (ICCPM) standard has emphasized for more than a decade (ICCPM, 2012). This is also an approach to projects and project management that distributed free/libre/software communities pioneered and elaborated throughout the past three decades. (Stallman, 1991) (Himanen & Torvalds, 2009) (IETF, 2010) (Potvin, 2014a)

Situating This Dissertation Amidst Recent Literature on Informatics in Project Management

> *This section presents and comments on excerpts from academic books and articles that have been published within the past three years (2019-2022), highlighting particular segments in bold font to indicate aspects shared with the design research detailed in the present dissertation.*

A comprehensive analysis of the impact of digital methods in project management practice and theory is provided in a 280-page book from Shaopei Lin and Dan Huang, published by Springer in 2020. They emphasize that the function of project management in not changing, but the methods and capabilities are: “We recognize that even though the emergence of [the] digital Internet ... will not totally change the basic principles of project management; however, the methodology, operation procedure and tools are changed.” (Lin & Huang, 2020, p. 261 They foresee that the proliferation of a distributed, decentralized Internet-enabled means of performing project management “will heavily influence ... the theoretical frameworks of project management, program management and portfolio management (p. 274), in particular:

Projects: real time data transfers accelerate process and efficiency;
Programs: innumerable factors can be reviewed, controlled and updated in parallel;
Portfolios*:* numerous initiatives managed with explicit criteria facilitates learning through comparison.

Lin and Huang explain that this transformation begins with platform and software “infrastructure building” which alters the basic framework for down-stream project teams, leading project management teams into iterative research and development cycles. Hence project managers who previously were a passive users of infrastructure, become active contributors to the emergent platforms that support their responsibilities. The highlighted segments in the excerpt below reflect the intended infrastructure design role of the present DWDS specification for “an Internet of Rules”:

> “The position change of individuals [in the] Internet era is quite obvious; ... their **social position will be changed from “company-staff” to “platform-individual”** ...and transform one’s working rule from “passive” to “active”, ... from [a] definite organization to **face the whole society for one’s contribution**. **The relationship between people [in the] Internet era will be no longer through ‘networking’, but through definite ‘rules’.** ... **The new generation of project management will be implemented on the Internet platform**, where the layout, planning, implementation, tools, control and operation will be carried out. ... [T]he subversive impact of Internet has seriously changed the ecology of project management. ... There is an irreversible trend that **the Internet will be used as a fundamental information platform for project management practice in each field**. (Lin & Huang, 2020, p. 250-251, emphasis added)

John McGrath and Jana Kostalova observe that the “digital transformation of project management approaches ... is not limited to just project documentation in electronic format” but it is “more about the digital transformation of the full project life cycle: initiation, planning, execution, monitoring and controlling and project closeout. (McGrath & Kostalova, 2020, p. 2)

> "Project management is experiencing a fundamental shift. The traditional skills of project management: delivering to schedule, budget and are no longer enough in the new world of constant change. Organizations are looking for a new skillset and competency: somebody who can drive organizational change and lead transform within the organization. ... While success of project management practices had commonly been attributed to the combination of tools, techniques and processes employed, more recent thinking has considered skills such as creativity, innovation as well as **faster decision making by empowered teams** to be the source of success." (McGrath & Kostalova, 2020, p. 7-8)

In a 2022 article entitled *“Digital project management: Rapid changes define new working environments”* Te Wu outlines a “combination of traditional project management and **the full adoption of digital tools and technologies in the management of projects**", that to say, not only projects with digital deliverables, but also core use of Internet and Web-based methods for distributed and mobile project management (Wu, 2022, p. 323).

Marina Filatova identifies additional aspects of the emerging project management discipline that also characterize the present DWDS design research undertaking:

> "[D]igitalisation in project management can create more effective **strategic collaboration among organizations** by creating a network. Digital platforms allow for real-time collaboration and communication, which increases the productivity and efficiency of project teams. I**nteractions within project teams leads to greater responsibility among team members and helps to create an environment that is conducive to collaboration. The digitalization of project management also integrates companies from different sectors of the economy,** and project managers become strategic leaders. Digital transformation enables managers to **apply modern technology in data analysis, to make decisions more quickly, easily and efficiently**, thereby increasing the impact of projects." (Filatova et al., 2021, 337, emphasis added)

In a 2020 paper entitled *"General and Specific: The Impact of Digital Transformation on Project Processes and Management Methods"* Alina Kozarkiewicz describes exactly the orientation to project management that has been intrinsic to the present DWDS design research, as indicated with bold text:

> [T]he most important aspects considered as the impact of DT on project management are as follows: 1) Access to data on the course of the project; 2) **IT tools supporting project management processes**; 3) Internal communication in project teams; 4) Project management methodology; 5) Greater customer orientation; 6) Process optimization. ... The traditional approach to project management, based on the use of methods of optimizing the course of the project over time, meeting deadlines as a key success factor or minimizing resources to ensure profitability, has been replaced by **an approach based on incremental product development, by a mobile, collaborative online team using modern IT tools, and dealing with data and experience as an important project resource.** (Kozarkiewicz, 2020, pp. 242-243, 246, emphasis added)

The 2019 paper by Jennifer Whyte which was cited at the beginning of this Appendix, entitled *"How Digital Information Transforms Project Delivery Models"*, further describes the type of product development of which DWDS is a genuine example, again highlighted with bold text:

> "There is a growing focus on digital **workflows to populate digital data sets and on how such data can be searched, accessed, tracked, reorganized, and analyzed using asset identifier codes**. Work to develop **industry-wide approaches to structuring digital information** was developed through all three industry/government initiatives studied, with **substantial work to develop and update standards**. ... digital information is changing what projects deliver, with **information becoming itself a deliverable**. Digital information is also changing how projects are delivered: **enabling greater sharing, remote access, searching, and updating of information with visibility across supply chains and with owners, operators, and end users**. ... This paradigm shift is **bringing new computational capability into project management**, with increasing use of statistics and search and pattern matching across vast datasets providing **new opportunities to transform project governance and project delivery processes**." (Whyte, 2019, p. 190-191, emphasis added)

In a more recent 2022 article *"Projects as interventions"*, Jennifer Whyte and Lara Mottee describe the pragmatist (pragmaticist) transdiscplinary methodology that has arisen in the past four years of published academic literature (Whyte et al., 2022) (Simpson & den Hond, 2022) (Thompson & Byrne, 2022) (Wenzel, 2022) (Wenzel et al., 2020). What they describe is, in its essence, a similar scope and methodology as the present author independently adopted throughout the past nine years of DWDS design research, but with conceptual and methodological roots in literature from the past century (James, 1922) (Maxwell, 1972) (Rosenthal & Bourgeois, 1977) (Feyerabend, 1982) (Kloppenberg, 1996) (Brandom, 2008) (Lalonde et al., 2010). The commonalities are highlighted in the excerpt from White and Mottee below:

> Understanding projects as interventions into nature implies a more inter-disciplinary approach, requiring new forms of scholarship, **bringing deep insights from the natural sciences and engineering together with those from business studies, data science and the social sciences**. ... Drawing on the American **Pragmatist tradition of theorizing**, the notion of **future making** articulates an approach to making futures through **reflective participatory practices**. (Whyte & Mottee, 2022, p. 935-936, emphasis added)

This mingling of natural sciences, engineering, data science, business studies, and social sciences is reflected also in a 2021 article by Flávio Copola Azenha et.al. entitled *"The Role and Characteristics of Hybrid Approaches to Project Management in the Development of Technology-Based Products and Services"*. Their hybrid 'project scope' is a close match to the present work, again as emphasized in the following:

> "Composed of a **long-term specification** with formal descriptions of the objectives and expected results for the project as a whole, and a short-term view for iterations, based on **metaphorical and abstract representations of each iteration objective**... Changes are identified and short-term planning is adjusted for each interaction, avoiding deviations in long-term planning. ... "the application of hybrid approaches to project management is recent from the academic point of view ... ideal for innovative projects, those involving a high amount of uncertainty and that cannot be undertaken without some level of planning, ...with a moderate degree of complexity and technical challenges." (Copola Azenha et al., 2021, p. 96-97, 105-106, emphasis added)

Bertha J. Ngereja and Bassam Hussein emphasized practical learning-by-doing in projects that cross organizational and sectoral boundaries, as indicated here:

> "This dimension of learning is denoted as l**earning between projects, inter-project learning, and cross-project learning**. Intra-project learning materializes when individuals are given the opportunity to experiment, reflect and accumulate knowledge individually or in groups while being engaged in a project. This is **primarily a learning-by-doing approach** and is a part of the experiential type of learning. ... the nature of the task/job [is] one of the preconditions for learning, that **has not been mentioned in the literature reviewed for this study**. (Ngereja & Hussein, 2021, p. 28, emphasis added)

A structure for inter-project learning-by-doing appears in a table of theoretical frameworks (columns) and operational practices (rows) provided by Serghei Floricel et.al in a 2014 article entitled *"Extending project management research".* They suggest that project management researchers "select from it ... to investigate certain types of theoretical and practical issues in project management" (Floricel et al., 2014, p. 1101). The present DWDS design research was completed prior my coming across this table, but in scope and style my present work can be situated in three particular cells in the column describing the "activity theory" of Yrjö Engeström (Engeström, 2009) (Engeström & Glăveanu, 2012), and in three rows drawn from Davide Nicolini's "five dimensions of practice" (Nicolini, 2012), namely: tools and databases; *c*ollaborative project work; and, ideas, procedures and formulas that persist beyond the current project (Floricel et al., 2014, p. 1102). In particular the present DWDS design research can be situated with the highlighted segments of the following excerpt on 'activity theory':

> “[A]ctivity theory is particularly useful for understanding the complex organizational forms observed in the New Economy, with **more fluid boundaries than traditional project organizations**. For example, online **communities involved in global open source software projects** pose new challenges in terms of creating, maintaining and sharing expertise. These distributed and heterogeneous settings can be analyzed as **networks of overlapping activity systems**. By doing so, project researchers can explain how **a group of people who have never met before can work together towards a common (or partially shared) “object.”** It is the idea of a common “object” that enables such temporary and distributed organization forms, by allowing shared conceptions of the activity. For project research in general, activity theory, in combination with the practice approach, can help grasp **the essence of temporary organizations, in particular of emerging ways of organizing work**. (Floricel et al., 2014, p. 1095, emphasis added)

Terminology: ‘Digital’ Versus ‘Automatic’, ‘Informatic’, ‘Algorithmic’ and ‘Electronic’

This afterword has considered the very recent proliferation of attention to ‘*digitalization’* and ‘*digital transformation’* in academic management journals, with a particular focus on the project management literature. However, what appears to be a new phenomenon is, in my assessment, really only tracking the use of newly-popularized but imprecise jargon to describe a shift that has been steadily underway for almost 80 years. Reflecting upon this has led me to remove such terms from the body of my own dissertation, except when referring to work of others using such terminology.

In the early 1940s John Atanasoff and Clifford Berry originated the method of using binary digits {0,1} to represent uncharged and charged capacitors in arrays, offering a simple method for scalable, adaptable, integrated storage and processing of the logic states {False, True} or {No, Yes} (Atanasoff, 1984) (Gustafson, 2000) (Grier, 2000). Their method came to be adapted in various ways throughout the entire informatics industry, but this digital machine-layer technique is never actually seen or directly engaged by end users. Generally when authors use some variant of the term ‘*digital’* they are actually referring to *automatic information processing.* This phrase appears as the subtitle of the seminal paper published in 1957 by Karl Steinbuch, entitled *Informatik: Automatische Informationsverarbeitung*. (Steinbuch, 1957) "Informatik is the scientific investigation and theory of information processing." (Steinbuch, 1958, p. 319) In the 1960s Peter Naur adapted this term as it became more widely used in reference to applied systems: "In the present context the term 'informatics' is used for the science and technology of information processing and the associated applications." (Naur, 1977, p. 5)

Section 4.2 of this dissertation reviews multiple logic data models using various two-, three-, and four-element sets. Section 4.4 also steps through diverse influences and inspirations spanning 70 years of electronically-automated information processing using programmable logic algorithms.

The so-called *digital transformation* has been underway for a very long time. Favoring precise terminology throughout this thesis, I confine use of the word *‘digital’* to circumstances where I am referring to binary, trinary and tetranary digital sets to express logic states, and I have edited this dissertation to use the terms ‘*automatic’*, ‘*informatic’*, *‘algorithmic’* and *‘electronic’* where each of these carries their precise meanings.

Appendix F: Excerpts from a Public Submission to a Regulatory Body

Response to FINRA's Request for Public Comment on the Future of Its Machine-Readable Rulebook Initiative

https://www.finra.org/rules-guidance/notices/special-notice-102122

Joseph Potvin

February 21, 2023

...

2. Our Perception of the Context and Purpose of FINRA's Machine-Readable Rulebook

The 850 "FINRA Rules" and thousands of interpretative texts, policy statements, change notices, and other guidance documents produced by the Financial Industry Regulatory Authority (FINRA), shape the self-regulated market operations of more than 3,500 brokerage firms and over 600,000 registered securities representatives in the United States.[91] FINRA's regulatory role is exercised under delegated authority from the Securities and Exchange Commission (SEC), in addition to the SEC's own legal framework of statutes (Securities; Securities Exchange; Trust Indenture; Sarbanes-Oxley; Dodd-Frank; JOBS) and operational rules, regulations, schedules and interpretations.[92] The member-funded not-for-profit FINRA, and the government-funded SEC statutory agency, jointly pursue securities market integrity and investor protection.

With operating revenues in excess of USD$1B annually, FINRA's activities include oversight services (consolidated rule-making, surveillance, examinations, fraud detection, enforcement, dispute resolution); membership (applications, registrations, training, certification, communication); and transparency services (automated reporting, advanced data analytics, verification audits, and formal examinations of securities firms based on risk, scale and scope of operations).[93] FINRA's in-house capability for informatics and data science involves over 500 software developers, who currently have more than 100 software applications under management. They enable the organization to maintain continuous surveillance of securities market activity, processing approximately 6 terabytes of data per day running hundreds of surveillance algorithms on an average of a billion financial transaction events to detect patterns that may signal market manipulation, insider trading and many other unfair activities in stock and bond markets. The empirical results of this scrutiny inform disciplinary actions such as censures, fines, suspensions, expulsions, and restitution to harmed investors.

In mid-2018 FINRA launched consultation and developmental work on a taxonomy for an eventual machine-readable rulebook. A two-level categorization of regulatory and industry terms was produced, with summary themes and a hierarchy of detailed topics. In Autumn 2022 FINRA then

91 FINRA was established in 2007 through consolidation of the member regulatory functions of the 'National Association of Securities Dealers LLC' and 'NYSE Regulation LLC', a subsidiary of the New York Stock Exchange. Through contracts it also took on responsibility for regulating the 'Nasdaq Stock Market', the 'American Stock Exchange', and the 'International Securities Exchange'. It would consolidate regulatory rules and enforcement, and operate utilities for trade reporting and essential over-the-counter operations.

92 https://www.sec.gov/about/laws/secrulesregs

93 In 2021 FINRA referred 758 fraud and insider trading cases for prosecution, suspended or barred 655 individual traders, expelled or suspended 4 brokerage firms, imposed $130M in fines, and ordered $47 in victim restitution.

launched two online prototype services with an initial set of the 40 most frequently viewed rules from full collection of 850. A Web application called FIRST (FINRA Rulebook Search Tool) provides an interface for users to locate FINRA rules through a step-wise selection of categories. And the FINRA API Platform (Application Programming Interface) facilitates automated keyword queries of the database of this sample of rules.

...

3. Constraints of FINRA's Current 'Machine-Readable' Rulebook, and Ways to Transcend Them

> ***What apparent constraints might prevent FINRA's Machine-Readable Rulebook initiative from meeting the requirements of its members' diverse conformance management approaches, and of its own market surveillance?***

The stated purpose of FINRA's "Machine-Readable Rulebook" is "to enhance firms' compliance efforts, reduce costs and aid in risk management". In our assessment there are three specific constraints inherent in FINRA's approach as currently described and prototyped. Left unresolved, these issues could prevent the initiative from accomplishing the objective of improved conformance management among industry members, and of enhancing its own market surveillance systems.

3.1 "Natural Language from Rule-Makers" Versus "Natural Language for Rule-Takers"

Recommendation

> *FINRA's "Machine-Readable Rulebook" initiative is designed to facilitate finding the regulations that securities brokers and dealers must conform with. Future work could include a systematic approach to providing auxiliary natural language summaries that would enhance their situational recall and understanding.*

Rationale

Rule makers who draft legislation, standards, interpretations and guidelines are, of course, obligated to express themselves with precision. They need to refine the wording of each rule to ensure that it states exactly what is intended. On the other hand, rule takers intent on rules conformance must perform complex situational recall of numerous obligations/exclusions, permissions/prohibitions, and encouragements/discouragements. Although securities dealers and brokers typically hold university degrees in finance, accounting, economics or business, and prepare for and pass exams to obtain and maintain their licenses, even the most intelligent and honest among them face the "precision-recall tradeoff" described half century ago by Cyril Cleverdon:

> "As a general rule it remains true that in a large number of situations, an improvement in recall can only be obtained with a loss in precision, or vice versa, and it is reasonable to operate a system using this as a working principle. However, the inverse relationship of recall and precision is not a fundamental law..." (Cleverdon, 1972, p. 195, 199)

Simplification for sophisticated professionals requires choosing terms and phrasing optimized for understanding and recall of the essentials. A commonly known illustration is the 200-word summary of the 2,500-word *Creative Commons Attribution 4.0 International License.* It is introduced with the caveat: "This is a human-readable summary of (and not a substitute for) the license." [94]

94 https://creativecommons.org/licenses/by/4.0/

Excerpt from the 'Data with Direction Specification'

"The Internet Engineering Task force specifies: " 'Simplification of language' here refers to ways of controlling expressions in a language to make reading or comprehension easier for particular target audiences". (Phillips & Davis, 2009) In the 1950s the UK Government had Ernest Gowers provide guidance in *Plain Words* for how to achieve straightforward communication, as this is indispensable to getting practical work done:

> "But what is this job that must be got on with? ... the writer's job is to make his reader apprehend his meaning readily and precisely. ... Even when he knows what he means, and says it in a way that is clear to him, is it always equally clear to his reader? If not, he has not been getting on with the job." (Gowers, 1954, p. 78)"

It was aeronautical engineer Clarence Johnson who emphasized "applying the simplest, most straight-forward methods possible to develop and produce new products" and then articulated the famous aphorism: "Keep it simple, stupid—KISS" (Rich, 1995, p 221, 231). System procedures, interfaces, and documentation, can benefit from the well-known 7±2 guideline that average human short-term memory capacity for processing information is constrained to about seven plus or minus two items (Miller, 1994), or its less prominent 4±1 refinement (Cowan, 2001) (Mathy & Feldman, 2012)." (Potvin 2023, p. 158, 75)

3.2 "Machine Readable" Versus "Machine Processable"

Recommendation

> *FINRA's "Machine-Readable Rulebook" initiative is premised on use cases where computing resources can support highly-expressive semantic data processing. Future work could include specialized support for speed-optimized, in-memory key-value sifting methods suited to algorithmic high-frequency transaction systems.*

Rationale

High speed, high volume data processing at the scale performed by algorithmic, electronic, automated and high-frequency trading systems, and by FINRA's market surveillance systems, need to validate rule conformance without being slowed down by compute-intensive parsing of expressive sentences or hierarchies of semantic tags.

Applying a meaningful taxonomy to natural language data is suitable for use cases within conventional "Semantic Web" scenarios in which local browsers or interactive apps have the small job to do of associating meaning with displayed texts while interacting with a human. But this method of semantic tagging of expressive natural language is not usable for extremely high-speed high-volume normative data processing. Even among some of the most advanced methods of interactive natural language, complex semantics have been replaced with brute-force stochastics. (Vaswani et al., 2017)

Excerpts from the 'Data with Direction Specification'

There are numerous techniques available for optimizing a rule system for speed and throughput. Following are section headings that identify various techniques employed in that particular design:

5.3 Methods for High Performance Decentralized Distributed Computing
5.3.1 Externalize Computational Work from Run-Time
5.3.2 Externalize Complexity from Expression with Simple Controlled Natural Language
5.3.3 Externalize Linguistic Complexity from Rule Structure, to Simplify Function
5.3.4 Externalize Engagement of Semantic Web Standards to Rule Makers and Rule Takers
5.3.5 Externalize Computability by Requiring Rule Expression to be NOT Turing-Complete
5.3.6 Externalize Control Data and Logical Relations Data by Separating Data from Procedure
5.3.7 Externalize the Data Processing Burden with Purposeful Structuring of Data Into Tables
5.3.8 Externalize Reusable Algorithms (In-Memory Retrieval of Cartesian Product Tables)
5.3.9 Externalize Declarative Conditions and Assertions from Logical Relations
(Potvin 2023, p. 155-200)

"[O]ptimal' rule systems ... enable individuals and entities to communicate normative propositions more *cost-efficiently* and *cost-effectively* than is otherwise currently feasible:

- *Cost Effectiveness:* Maximize the quality of direction-intrinsic data communication within a given amount of time, resources and risk.
- *Cost Efficiency:* Minimize the time, resources and risk needed to achieve an intended quality of direction-intrinsic data communication." (Potvin 2023, p. 102)

3.3 "Rule Book" Versus "Rule System"

Recommendation

> *FINRA's "Machine-Readable Rulebook" initiative involves delivery of two online services: Web-based rules search, and an API for rulesbase queries. Future work could include free/libre/open collaborative experimentation with end-to-end systems to advance the normative performance of the US securities market.*

Rationale

The statistics in the "Regulatory Actions and Corporate Financing Review 2017–2021" online at https://www.finra.org/media-center/statistics are worth some reflection. In those five years the number of investor complaints received by FINRA has nearly quintupled, and yet the number of disciplinary actions filed, and the number of individuals barred and suspended, each declined by almost half. No interpretation of these apparently contradictory trends (greater rules conformance, yet lower investor protection?) is provided in FINRA's 2021 Annual Report.[95] Perhaps there were fewer violations of the rules overall, but those transgressions which did occur affected many more investors, more severely.

A whole systems perspective on FINRA's "Machine-Readable Rulebook" initiative considers the general trends and dynamic forces shaping rules communication, surveillance, response and

95 https://www.finra.org/sites/default/files/2022-06/2021-FINRA-Financial-Annual-Report.pdf

outcome. As a dynamic interactive phenomenon, FINRA inevitably faces *"The Problem-Solvers' Paradox":* the greater and more sustained FINRA's success in terms of rules conformance, the lower the perceived need for its services, which can weaken vigilance and increase vulnerability to fewer but more severe abuses. A systems designer considers ways to re-frame this dynamic, for example one might brainstorm a *Market Integrity Index Fund* that would increase in value as verifiable normative performance indicators demonstrate improvements in both rules conformance and investor protection.

Excerpt from the 'Data with Direction Specification'

This design research provides a rationale, a functional specification and partial prototype working components to solve the following general class of problem:

> *Agent A, interacting with Agent B, requires knowledge of one or more externally-managed rules from Agents C..n that are 'in effect' for given contexts, and are 'applicable' to a set of event categories, and are 'invoked' by particular circumstances, where:*
>
> *(i) A and B may or may not know about C..n's rules, or about any updates to them, but either or both would prefer to obtain all available facts about relevant rules when interacting.*
>
> *(ii) C..n may or may not know about A and B in particular, nor about their particular medium of interaction, but can expect A or B or their medium of interaction to be capable of exchanging data with a generic medium common to A..n.*
>
> *(iii) A and B would tolerate the risk of exposing limited data through the generic medium so that it can be used to select information about relevant rules from C..n.*

...The "Data With Direction Specification" (DWDS) describes a type of distributed, general purpose system that individuals and organizations can use to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate rules across any informatics network with precision, simplicity, scale, speed, resilience, and deference to prerogative. DWDS describes a class of data-processing pipeline with the underlying relation: 'IS + RULE ⟹ OUGHT'. (Potvin 2023, p. 57, 146)

4. Our Perspective on the Potential of FINRA's 'Machine-Readable' Rulebook

> ***How might FINRA's Machine-Readable Rulebook be adapted to improve human comprehension and recall of the rules; to meet the speed and volume requirements of algorithmic transactions; and to reduce the rules management burden?***

The Xalgorithms community perspective on rule systems design is detailed in a recently-completed 250-page thesis, which we include as supporting documentation to the present submission. The GitLab URL provided below supplies the most recently edited version, and an overview presentation deck.

Potvin, J. (2023). *Data with Direction: Design Research Leading to a System Specification for 'an Internet of Rules'. Dissertation in partial fulfillment of a Doctorate in Business Administration—DBA (Project Management).* Université du Québec—Outaouais Campus (UQO)]. *License CC-by 4.0*
https://gitlab.com/xalgorithms-alliance/data-with-direction-specification/dwds-documents/-/tree/master/current

The following three sections highlight elements of how FINRA's "Machine-Readable Rulebook" can be integrated with the DWDS "Internet of Rules" concept and functional design to advance conformance management and investor protection through improved human access to, as well as comprehension and recall of the rules; to facilitate high performance operationalization of FINRA's rules in algorithmic transaction and surveillance systems; and to reduce FINRA's internal rules maintenance workload.

4.1 Situating FINRA's Rulebook in the DWDS 'Internet of Rules' System Concepts and Functions

Figure 54: A View of FINRA, its Rulebook, and its Members in the Conceptual Space of an End-to-End Rules System. Adapted from: (Potvin 2023, Fig. 10, p. 148)

	Members	**Rulebook**	**FINRA**
NORMATIVE DATA *MUST, MAY and SHOULD*	**Empirical**	**Declarative**	**Imperative**
Informational **Data, Metadata, Schema**	***Fact:*** ***An Event / A Status Change*** (generated / reported / detected) (pending / estimated / potential)	***Communication:*** ***Best Available Information*** (accessible and verifiable) (shaped by relationships)	***Prerogative:*** ***Social or Institutional Agency*** (authority / agreement / preference) (subsidiarity / paramountcy)
Operational *"DWDS"*	[is.dwd] request → ***RuleTaker*** Role ← [ought1.dwd] response	***RuleReserve*** Network	← [rule.dwd] & [lookup.dwd] ***RuleMaker*** Role patterns →
Contextual	***Normative Circumstance*** A set of primary facts invoke some normative propositions, and thus establish a normative circumstance.	***Normative System*** There exists an ensemble of rules which characterize a particular normative order.	***Normative Assertion*** A requirement includes one or more normative propositions.
Practical **MUST, MAY, SHOULD** and their synonyms	***Normative Fact*** or ***Ruled-Based Fact*** A set of primary facts invokes a normative proposition, and therefore establishes the existence of a normative fact.	***Normative Proposition*** or ***Rule Documentation*** There exists a normative proposition relevant to this data which is 'in effect' for this context, and 'applicable' to these facts.	***Norm*** or ***Rule*** Institutional or social norms for *practical* action or status are 'in effect' for a context, and 'applicable' to foreseen facts.
Ethical **MUST, MAY, SHOULD** and their synonyms	***Deontic Fact*** A set of primary facts invokes a normative proposition based on utility, logic, ethics or aesthetics, and thus establishes the existence of a deontic fact.	***Deontic Proposition*** There exists a normative proposition based on utility, logic, ethics or aesthetics 'in effect' for this context, and 'applicable' to these facts.	***Deontic Rule*** Institutional or social views for *ethical* action or status are 'in effect' for a context, and 'applicable' to foreseen facts.

Figure 55: The Functional Role of FINRA in the "Rule Maker Role" of the DWDS Sequence Diagram. Adapted from: (Potvin 2023, Fig. 11, p. 150)

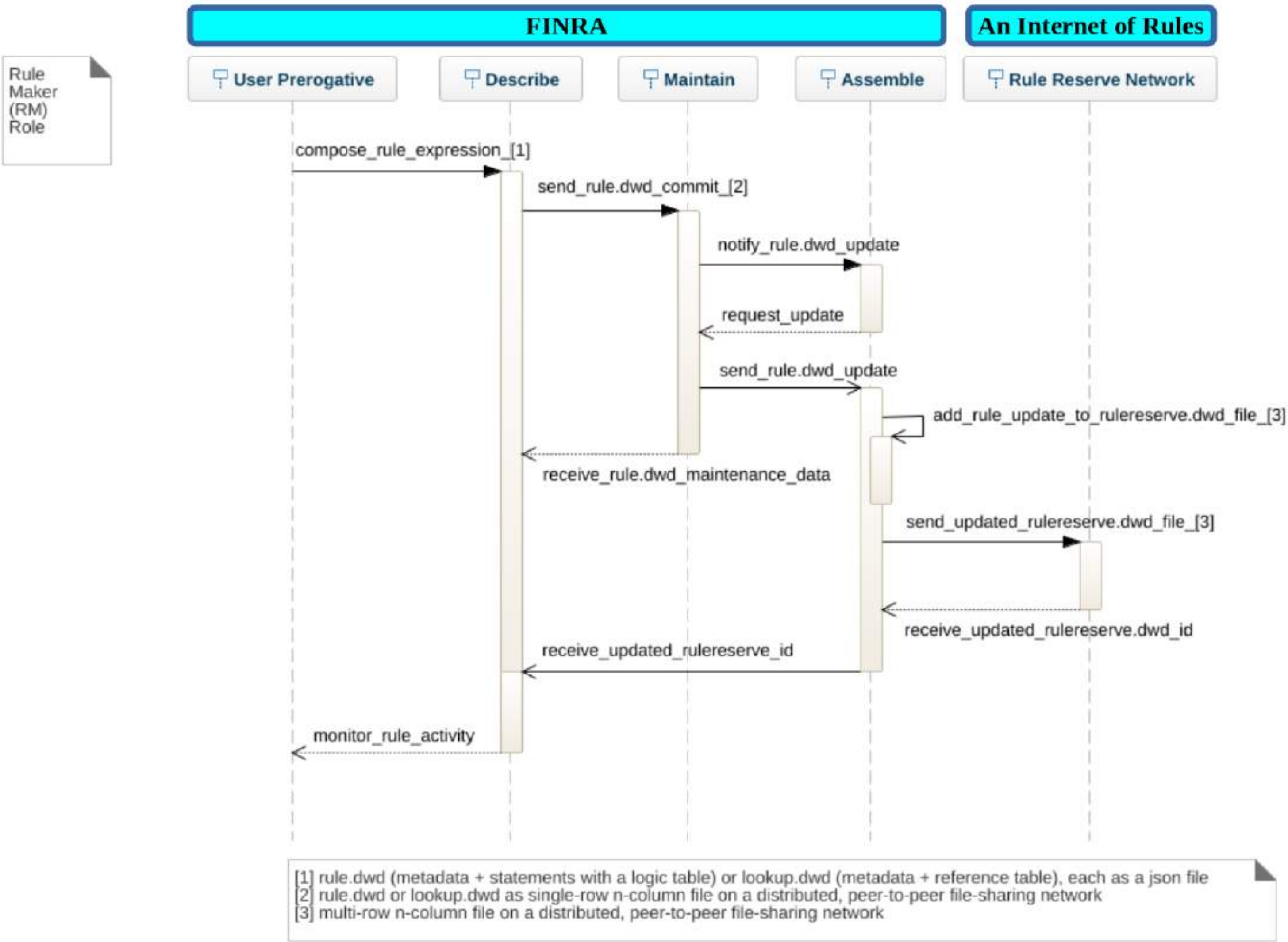


Figure 56: The Functional Role of the FINRA's Rulebook in the "Subset Rule Reserve Role" of the DWDS Sequence Diagram. Note that this only shows the top half of the Rule Reserve Network functions. Adapted from: (Potvin 2023, Fig. 12, p. 151)

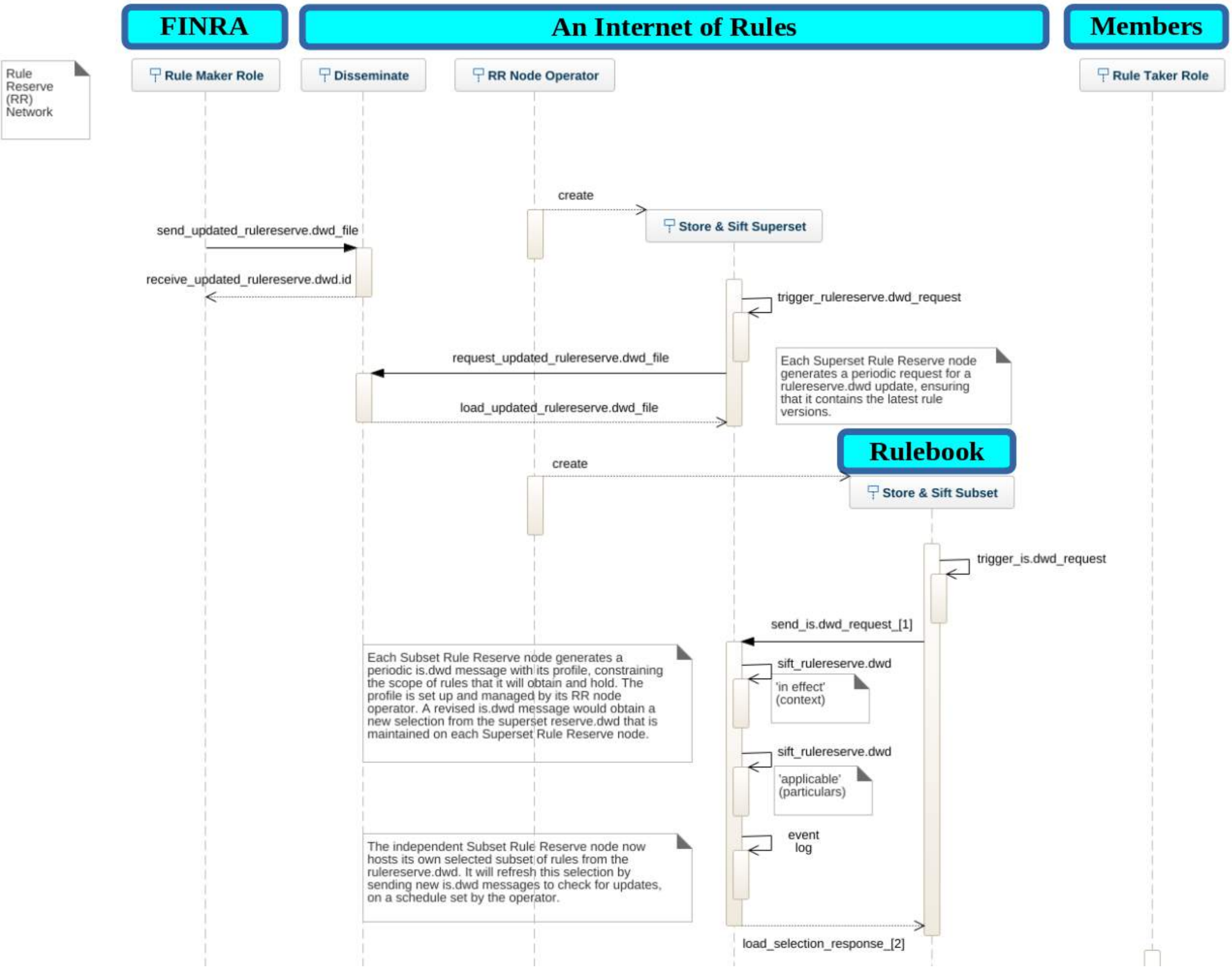
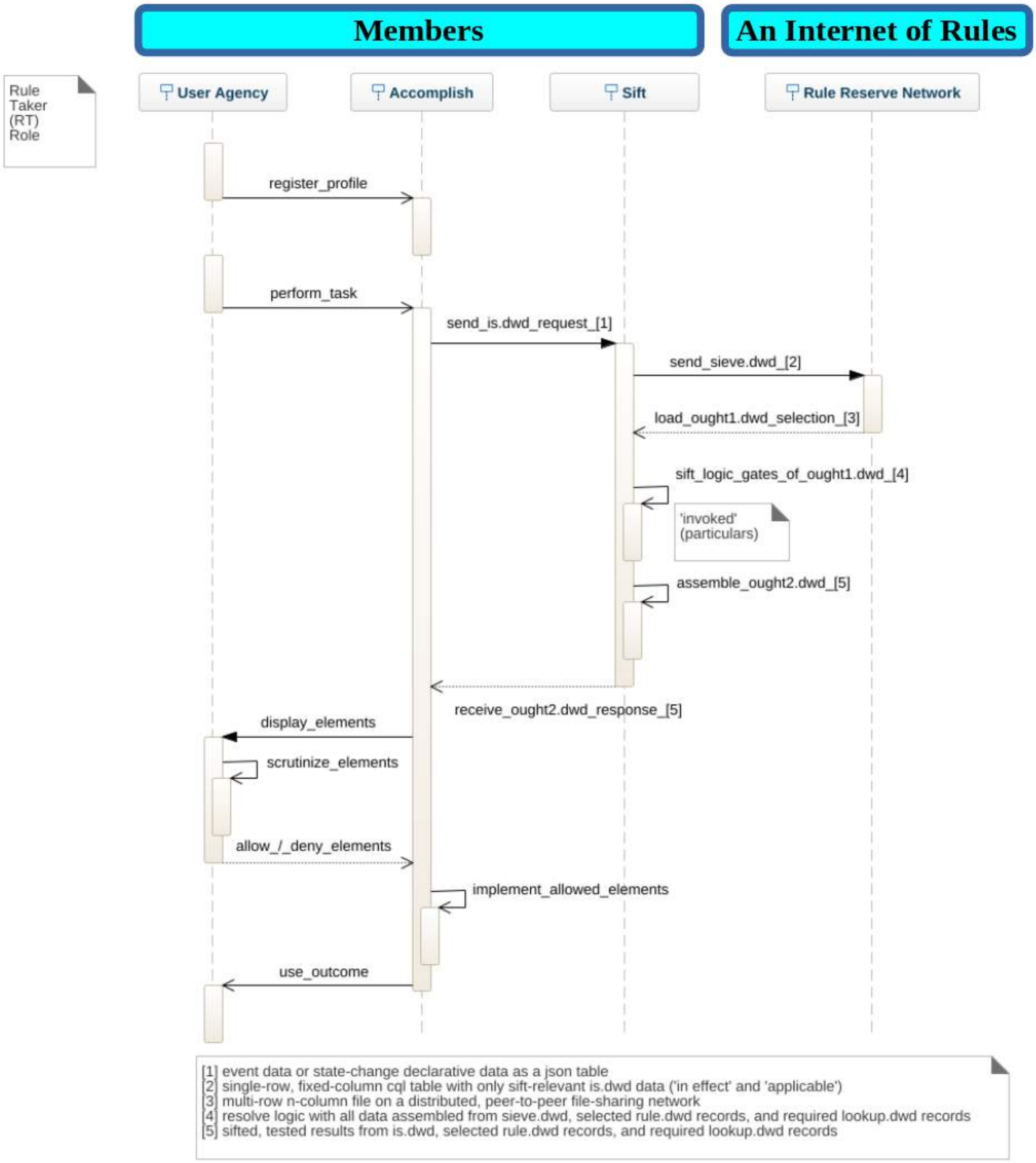


Figure 57: The Functional Role of FINRA Member Firms and Individuals in the "Rule Taker Role" of DWDS Sequence Diagram. Adapted from: (Potvin 2023, Fig. 13, p. 152



4.2 Transcribing Two Sample FINRA Rules to DWDS RuleData Form

For this submission two sample FINRA rules (3240; 4140) were structured into DWDS RuleData form. The corresponding JSON files are attached as supporting documentation, and working views of their respective logic gates are provided on the following pages, as portrayed in the graphical user interface of the RuleMaker Web app. The images are also attached separately to facilitate reading of the small text of the sentences that comprise the Input Conditions and Output Assertions.

The reader should take into account the following considerations:

(a) The permutation scenarios are to be read vertically, as Scenario A, Scenario B, and so on.

(b) The symbols have the following meanings:

Figure 58: Meanings of Symbols in DWDS Logic Gates. Adapted from (Potvin 2013, Fig 19, p. 192)

Input Symbol	Input Conditions	Output Symbol	Output Assertions
X	NO	⊘	NOT
✓	YES	!!	MUST
&	YES AND NO	O	MAY
?	YES OR NO	!	SHOULD

(c) The sentences of the Input Conditions and Output Assertions were adapted in three ways:

i. As discussed in Section 3.1 of this submission, the natural language of rule-makers is not necessarily the same as natural language suited to rule-takers. The metadata for each rule include the URL to the original regulatory text so users can readily consult the original.

ii. Each sentence is ‘shoehorned’ into DWDS “finite state grammar” of 6 syntactic elements;

iii. Each sentence has been adapted to the RuleSpeak guidelines to the extent practical within the constraints of the DWDS six syntactic elements (e.g. Use simple language; Break rules into atomic sentences; Avoid ambiguities; etc…). (Ross, 2023) (OMG, 2016)

iv. To experiment with a potential time-saving method, I instructed a vanilla (GPT)[96] as follows:

```
Re-write the following rule using only discrete declarative
sentences in a style that conforms with the essential
practices of "RuleSpeak", starting a new line for each
sentence, and without leaving out any operational details
or references. "4140. Audit. (a) FINRA may at any time...
```

This successfully transformed the original text of the sample regulation into well-structured declarative sentences. To further expedite the process of inserting the sentences into the six syntactic elements of DWDS RuleData I instructed the GTP with this:

```
Identify the 'subject', the 'predicate', and the 'object'
in each of the following sentences...
```

This saved some time, but with mediocre results. Probably a GTP could be trained with a set of declarative sentences pre-partitioned as described, to obtain higher accuracy.

96 OpenAI/GPT-3 https://platform.openai.com/playground

This representation of the logic structure of a rule in the DWDS RuleMaker Web app facilitates discussion and refinement of the individual sentences for the Input Conditions and Output Assertions, and well-organized consideration of the potential permutations to be anticipated in the logic relations.

Figure 59: Version 0.1.0 of a DWDS Logic Gate for FINRA Rule 3240: "Borrowing From or Lending to Customers", as seen in the RuleMaker interface.

The RuleMaker working environment provides for both “machine readable” and “machine processable” rule expression. The JSON record of this logic gate, which includes the rule metadata and optional descriptive fields, is auto-generated by the RuleMaker Web app. It can saved to one’s local drive and/or published to the Internet (IPFS) on any node of the RuleReserve Network.

This entire rules management process is in the hands of subject matter experts, while software programmers are focused on ensuring that the enabling applications are working properly. There is no requirement for software programmers to interpret regulation semantics or the rule logic.

In the examples provided here in Figures 59 and 60, the sentences have not yet been aligned to FINRA’s semantic taxonomy. That work requires more familiarity with the particular controlled natural language schema than the present author currently possesses. However this version 0.1.0 provides a convenient venue for collaboration to do so. FINRA’s taxonomy would be applied to the sentence elements. Discussion is required to determine exactly how this should be performed and displayed in RuleMaker.

Figure 60: Version 0.1.0 of a DWDS Logic Gate for FINRA Rule 4140: "Audit", as seen in the RuleMaker interface.

Once all the metadata, logic gate data, and optional descriptive data are entered into the RuleMaker Web app for a rule, the user can have it automatically generate the JSON file for local storage and/or Internet publication to IPFS, which supplies a unique Content Identifier (CID) for that precise version of that rule. Below is part of the JSON record for Rule 4140, from Figure 60.


```
{
    "id": "ce4c3fa7-1c84-4f00-8cc5-dfc11eee947c",
    "uuid": "ce4c3fa7-1c84-4f00-8cc5-dfc11eee947c",
    "rule_id": "ce4c3fa7-1c84-4f00-8cc5-dfc11eee947c",
    "rulereserve_nodes": "*",
    "version_standard_url": "https://semver.org/",
    "dwds_schema_version": "0.0.0",
    "properties": {
        "id": "ce4c3fa7-1c84-4f00-8cc5-dfc11eee947c"
    },
    "metadata": {
        "rule": {
            "120_title": "4140. Audit",
               "240_summary": "FINRA Rules\n4000. FINANCIAL AND OPERATIONAL RULES\n4100. FINANCIAL CONDITION\n4140. Audit",
             "960_explanation": "FINRA may require any member to have an audit or examination of its accounts conducted by an independent public accountant. The audit or examination must follow attestation, review, and consultation standards specified by the AICPA and any additional requirements set by FINRA. The audit or examination is directed by FINRA's Executive Vice President in charge of financial responsibility, or a delegate of theirs. Any member who does not file the relevant audited financial and/or operational report or examination report within the given timeframe will be subject to a late fee listed in Schedule A Section 4(g)(1) of the FINRA By-Laws.",
            "version": "0.1.0",
            "criticality": "experimental",
            "url": "https://www.finra.org/rules-guidance/rulebooks/finra-rules/4140",
            "pattern": "",
            "pattern_version": "",
            "rulemaker_entity": [
                {
                    "name": "Financial Industry Regulatory Authority (FINRA)",
                    "url": "https://www.finra.org/",
                    "uuid": "027d7f9e-0e7d-44cd-a893-5c12c6a20d0b"
                }
            ],
            "rulemaker_manager": [
                {
                    "name": "Xxxx",
                    "email": "Xxxx@finra.org",
                    "contact": "General Counsel, Office of the General Counsel",
                    "uuid": "b88e561c-2d5c-464b-a88a-de66f45096f0"
                }
            ],
            "rulemaker_author": [
                {
                    "name": "Joseph Potvin",
                    "email": "jpotvin@xalgorithms.org",
                    "contact": "",
                    "uuid": "49354a2a-fad3-44f8-bb0d-3891e6ed0d34"
                }
            ],
            "rulemaker_maintainer": [
                {
                    "name": "Joseph Potvin",
                    "email": "jpotvin@xalgorithms.org",
                    "contact": "",
                    "uuid": "4f2e65f7-e22e-40a7-a705-b3e89c8226fb"
                }
            ]
        }
    },
    "in_effect": [
        {
            "country": "US",
            "subcountry": "",
            "timezone": {
                "start": "UTC-05:00",
                "end": "UTC-05:00"
            },
            "start": "1992-08-12T04:00:01.000Z",
            "end": "2011-08-12T04:59:59.000Z"
        }
    ],
```

```
    "category_applicable": {
        "industry": [
            {
                "isic_code": "6611",
                "isic_name": "Administration of financial markets"
            }
        ],
        "good_service_asset": [
            {
                "unspsc_code": "64110000",
                "unspsc_name": "Securities"
            }
        ]
    },
    "data_sources": [],
    "input_conditions": [
        {
            "sentence": [
                {
                    "determiner": "This"
                },
                {
                    "noun": "entity"
                },
                {
                    "predicate_verb": "is"
                },
                {
                    "description": "a registered member"
                },
                {
                    "attribute": "of FINRA (FINRA member entity),"
                },
                {
                    "past_participle_verb": "as validated in the FINRA membership registry."
                }
            ],
            "scenarios": {
                "A": "01",
                "B": "01",
                "C": "01",
                "D": "01",
                "E": "01",
                "F": "01",
                "G": "01",
                "H": "01"
            }
        },
        {
            "sentence": [
                {
                    "determiner": "This"
                },
                {
                    "noun": "FINRA member entity"
                },
                {
                    "predicate_verb": "has been instructed to file"
                },
                {
                    "attribute": "within a given time frame,"
                },
                {
                    "description": "an audited financial and/or operational report or examination 
report to validate the accuracy or integrity of its financial statements, books and records or prior 
audited financial statements,"
                },
                {
                    "past_participle_verb": "as instructed."
                }
            ],
            "scenarios": {
                "A": "00",
                "B": "01",
                "C": "01",
                "D": "01",
                "E": "01",
                "F": "01",
                "G": "01",
                "H": "01"
            }
        },  ...
```

(The JSON representation of the logic gate continues, followed by optional descriptive data.)

4.3 Draft Charter for a Financial Securities Regulations Working Group

In Section 1 of this submission we explained that Working Groups hosted by Xalgorithms Foundation have their own written charter, managing their own donated funds. Participants can include businesses, governments, academics and not-for-profits, civil society communities and individuals, collaborating under Xalgorithm's 100% free/libre/open source model based on the Apache 2.0 license (RuleMaker Web app and RuleTaker embedded component) and the AGPL 3.0 license (RuleReserve network service), while documentation and data are shared under the CC-by 4.0 International license. One or more "Contributor Agreements" can be appended to a charter, and tailored to circumstance.

Following is 'first draft' working text towards the potential charter for a "Financial Securities Regulations Working Group".

Draft for Discussion

<u>Issue to be Addressed:</u> Financial Securities Market Integrity and Investor Protection

A well-functioning financial securities market operates on sets of rules and a cost-effective, cost-efficient generic rules system. Market integrity depends on human accessibility, comprehension and recall of those rules, and on high performance operationalization of the rules in algorithmic transaction and surveillance systems.

<u>Requirement:</u> On-Demand Delivery of Regulations 'In-Effect', 'Applicable' and 'Invoked'

Financial securities dealers and regulators have a common interest in event-triggered transmission of concise, current, and correct information about normative rules that are: 'in effect' for given dates/times, identities and jurisdictions; 'applicable' to a set of industry and product/service categories; and, 'invoked' by particular event circumstances; in a manner that is readily comprehensible to humans and directly usable in high-performance applications and platforms.

The behavioural and operational aspects of financial securities regulations are far more likely to be understood and conformed with when simple human-readable and fast machine-processable assertions of MUST, MAY and SHOULD (or their syonymns or negatives) are delivered on-demand to individuals, organizations and/or their machines at the instant they are relevant.

<u>Proposed Approach:</u> The Data With Direction Specification (DWDS) for "an Internet of Rules"

The "Data With Direction Specification" (DWDS) operationalizes the essential conceptual relation:

'IS + RULE $\Longrightarrow$ OUGHT'

The specification describes a type of distributed, decentralized, general purpose end-to-end data-processing pipeline that individuals and organizations can use to author, publish, discover, fetch, scrutinize, prioritize and, with agreement of direct stakeholders, automate rules across any informatics network with precision, simplicity, scale, speed, resilience, and deference to prerogatives, agreements and preferences. The functional design involves a RuleData data structure suitable for any platform and any language, a RuleMaker application with the *imperative* role in normative communication (i.e issuing rules), a RuleReserve network service with the *declarative* role (identifying rules that are 'in effect' for a context and 'applicable' to a set of categories), and a RuleTaker component with the *empirical* role (sending a set of circumstantial facts and receiving facts about rules deemed to be invoked by the those facts). Operated together these give rise to an "Internet of Rules" – a method by which independent, self-contained rules are transmitted efficiently and flexibly from the source repositories in which they are maintained, to the applications that use them.

One-Year Workplan: April 2023 to March 2024

Following is a tentative one-year project schedule oriented to the delivery of interim results and management check-points. This serves as a guide only, to be updated as determined by participants.

- **Month 1:**
  - Use the present one-year Working Group plan to elaborate particular objectives for various community contributors, and to frame the relationships with stakeholders.
  - Create an effective participatory R&D collaborative trajectory involving participants from multiple data supply organizations.
  - Identify a sample of rules for community testing, that range from simple to complicated.
  - Design structured test protocols for RuleTaker implementations in at least three widely deployed production-class algorithmic trading systems currently in use for securities.
  - Adapt or create a basic online test service for validating automated rule conformance:
    - Multilingual, accessible (WCAG 2.0) end-user interface.
    - Rapid iterative diagnosis and documentation of discrepancies.
    - Comprehensive task management workflow.
- **Months 2-3-4:**
  - Test transaction scenaros with RuleMaker, RuleReserve, RuleTaker reference implementations.
  - Incrementally increase rule complexity; refining the process for accuracy and for speed.
  - Refine the online service for validating rule conformance.
  - Develop a draft risk management model of “Internet of Rules” users.
  - Jointly develop and present a first interim report to stakeholders.
- **Months 5-6-7:**
  - Incrementally broaden collaborative work on rule expression and validation.
  - Roll out and support version 1.0 of the online service.
  - Publish version 1.0 documentation (technical, financial, legal).
  - Broaden consultations (technical, financial, legal).
  - Create hypothetical management/financial models for proliferation.
  - Jointly develop and present a second interim report to stakeholders.
- **Months 8-9-10:**
  - Increase collaborative work on rule expression and validation.
  - Test and debug complicated rules, exceptions, anomalies and dependency chains (forward-chained, backward-chained rules).
  - Commence scheduled version updates for each quarter (3 months).
  - Refine and publish documentation (technical, financial, legal).
  - Jointly develop and present a third interim report to stakeholders.
- **Month 12**
  - Develop for discussion and refine a workplan for Year 2.
  - Contract out an arms-length study for stakeholder/community views.
  - Assess demand for training, and make arrangements accordingly.
  - Assess demand for support, and make arrangements accordingly.

Cleverdon, C. W. (1972). On the Inverse Relationship of Recall and Precision. Journal of Documentation, 28(3), 195–201. https://doi.org/10.1108/eb026538

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. Behavioral and Brain Sciences, 24(1), 87–114. https://doi.org/10.1017/S0140525X01003922

Gowers, E. (1954). The Complete Plain Words. UK Stationary Office.

Mathy, F., & Feldman, J. (2012). What's magic about magic numbers? Chunking and data compression in short-term memory. Cognition, 122(3), 346–362. https://doi.org/10.1016/j.cognition.2011.11.003

Miller, G. A. (1994). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review, 101(2), 345–352.

OMG. (2016). The RuleSpeak Business Rule Notation. Semantics of Business Vocabulary and Business Rules (SBVR), v1.4. Object Management Group. https://www.omg.org/cgi-bin/doc?formal/19-10-07.pdf

Phillips, A., & Davis, M. (2009). Tags for Identifying Languages (Request for Comments RFC 5646). Internet Engineering Task Force. https://doi.org/10.17487/RFC5646

Potvin, J. (2023). Data with Direction: Design Research Leading to a System Specification for 'an Internet of Rules'. Submitted to the Board of Examiners in partial fulfillment of a Doctorate in Business Administration—DBA (Project Management). License CC-by 4.0 [Université du Québec—Outaouais Campus (UQO)]. https://gitlab.com/xalgorithms-alliance/data-with-direction-specification/dwds-documents/-/tree/master/current

Rich, B. (1995). Clarence Leonard (Kelly) Johnson (1910-1990): A biographical memoir. National Academy Press. http://www.nasonline.org/publications/biographical-memoirs/memoir-pdfs/johnson-clarence.pdf

Ross, R. (2023). Rules: Shaping Behavior and Knowledge (1st edition). Business Rule Solutions, LLC. https://www.brsolutions.com/rules-shaping-behavior-and-knowledge-book.html?src=BRC

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need (arXiv:1706.03762). arXiv. http://arxiv.org/abs/1706.03762

Reference List

Abdulsalam, S., Zong, Z., Gu, Q., & Meikang Qiu. (2015). Using the Greenup, Powerup, and Speedup metrics to evaluate software energy efficiency. *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*, 1–8. https://doi.org/10.1109/IGCC.2015.7393699

Abelson, H., Sussman, G., & Sussman, J. (1984). *Structure and interpretation of computer programs, (Second Edition, 2016)* (Vol. 33). MIT Press. http://web.mit.edu/alexmv/6.037/sicp.pdf

Adams, G., & Balfour, D. (2015). *Unmasking Administrative Evil*. M.E. Sharpe.

Adleman, L. (1994). Molecular computation of solutions to combinatorial problems. *Science*, *266*(5187), 1021–1024. https://doi.org/10.1126/science.7973651

Alchourrón, C. (1969). Logic of norms and logic of normative propositions. *Logique et Analyse*, *12*, 242–268.

Aleksic, S. (2013). Energy, Entropy and Exergy in Communication Networks. *Entropy*, *15*(12), 4484–4503. https://doi.org/10.3390/e15104484

Alexander, C. (1979). *The Timeless Way of Building*. Oxford University Press.

Alfieri, A., Becker, R., & Havinga, I. (2007). *What defines an international statistical standard and other types of international statistical publications in economic statistics?* UNCEEA. https://unstats.un.org/unsd/envaccounting/ceea/meetings/UNCEEA_2_6.pdf

Allen, T. (1987). Hierarchical complexity in ecology: A non-euclidean conception of the data space. *Vegetation*, *69*, 17–25.

Allen, T. F. H., & Hoekstra, T. W. (1990). The confusion between scale-defined levels and conventional levels of organization in ecology. *Journal of Vegetation Science*, *1*(1), 5–12. https://doi.org/10.2307/3236048

Alvesson, M., & Spicer, A. (2012). A Stupidity-Based Theory of Organizations. *Journal of Management Studies*, *49*(7), 1194–1220. https://doi.org/10.1111/j.1467-6486.2012.01072.x

Amin, A., & Mridha, M. (2020). A Mini View of PLC. *International Journal of Research in Advanced Engineering and Technology*, *6*(2), 23–25.

Anderson, A. R., & Belnap, N. D. (1975a). *Entailment: The logic of relevance and necessity* (Vol. 2). Princeton University Press.

Anderson, A. R., & Belnap, N. D. (1975b). *Entailment: The logic of relevance and necessity* (2nd print. with corrections, Vol. 1). Princeton University Press.

Anderson, C. (2015). *Creating a Data-Driven Organization: Practical Advice from the Trenches*. O’Reilly Media, Inc.

Andrew, A. M. (2012). Cloud computing: Views on Cybersyn. *Kybernetes*, *41*(9), 1396–1399. https://doi.org/10.1108/03684921211275450

Anellis, I. H. (2012). Peirce’s Truth-functional Analysis and the Origin of the Truth Table. *History and Philosophy of Logic*, *33*(1), 87–97. https://doi.org/10.1080/01445340.2011.621702

Ansar, A., Madrigal, D., & Collins, S. (2019). Scaling Clean Energy for Data Centres: Trends, Problems, Solutions. In M. Ozawa (Ed.), *In Search of Good Energy Policy* (pp. 202–223). Cambridge University Press. https://doi.org/10.1017/9781108639439.015

Anscombe, G. E. M. (1957). *Intention (Reprinted in 2000)*. Harvard University Press.

Antonides, J. (2022). *A Cognition-based Analysis of Undergraduate Students' Reasoning about the Enumeration of Permutations (Preview)* [PhD (Teaching and Learning), Ohio State University]. https://www.proquest.com/openview/87e3564874cf14fdf0f4a6ac4fd17b8f/1?pq-origsite=gscholar&cbl=18750&diss=y

Archer, M. (2007). *Making our way through the world: Human reflexivity and social mobility*. Cambridge University Press, UK.

Archer, N., & Ghasemzadeh, F. (1999). An integrated framework for project portfolio selection. *International Journal of Project Management*, *17*(4), 207–216. https://doi.org/10.1016/S0263-7863(98)00032-5

Aristotle. (1932). *Politics* (H. (Harris) Rackham, Trans.). London : Heinemann. http://archive.org/details/politicsrackh00arisuoft

Aristotle. (1991). *On rhetoric: A theory of civic discourse* (G. A. Kennedy, Trans.). Oxford University Press.

Arkin, A., & Endy, D. (1999). *A Standard Parts List for Biological Circuitry*. University of California (Department of Bioengineering and Chemistry); Lawrence Berkeley National Laboratory (Physical Biosciences Division); and the Molecular Sciences Institute. http://dspace.mit.edu/bitstream/1721.1/29794/1/Arkin.Endy.DARPA.pdf

Arntz, M., Gregory, T., & Zielinski, U. (2016). *The Risk of Automation for Jobs in OECD Countries: A Comparative Analysis* (OECD Social, Employment and Migration Working Papers No. 189; OECD Social, Employment and Migration Working Papers, Vol. 189). https://doi.org/10.1787/5jlz9h56dvq7-en

Arrow, K. (1984). The value of and demand for information. In *The Economics of Information: Collected Papers of Kenneth J. Arrow* (pp. 106–114). Basil Blackwell.

Atanasoff, J. (1984). Advent of Electronic Digital Computing. *IEEE Annals of the History of Computing*, *6*(3), 229–282. https://doi.org/10.1109/MAHC.1984.10028

Atkinson, C. (2018a, April 26). Disruptive trade technologies will usher in the 'internet of rules.' *LSE Business Review*. https://blogs.lse.ac.uk/businessreview/2018/04/26/disruptive-trade-technologies-will-usher-in-the-internet-of-rules/

Atkinson, C. (2018b). *An Internet of Rules for all: Can "Trade Policy 3.0" foster inclusive trade? In: Trade for Development News (Online Journal) published by the Enhanced Integrated Framework (EIF), through the Executive Secretariat at the World Trade Organization*. WTO. https://trade4devnews.enhancedif.org/en/op-ed/trade-policy-3-0-inclusive-trade

Atkinson, C. (2018c, July 12). What is 'trade policy 3.0'? *World Economic Forum*. https://www.weforum.org/agenda/2018/07/trade-policy-3-0-will-foster-inclusive-trade/

Atkinson, C., & Potvin, J. (2022a). Algorithmic Governance for Resilient Trade and Development in Africa. In F. Olayele & Y. Samy (Eds.), *Unpacking Trade and Development Financing in Africa: Rethinking Resource Mobilization Pathways*. Routledge.

Atkinson, C., & Potvin, J. (2022b). Implementing the African Continental Free Trade Area: A Simple, Scalable, and Fast Computational Approach for Algorithmic Governance. In F. Olayele & Y. Samy (Eds.), *Sustainable Development in Post-Pandemic Africa: Effective Strategies for Resource Mobilization*. Routledge. https://www.routledge.com/Sustainable-Development-in-Post-Pandemic-Africa-Effective-Strategies-for/Olayele-Samy/p/book/9781032027609

Atkinson, C., & Schubert, N. (2021). Augmenting MSME Participation in Trade with Policy

Digitalisation Efforts: Chile’s Contribution to an “Internet of Rules.” *Trade, Law and Development*, *13*(1), 1–20.

Avelhan, B. L., & Zylbersztajn, D. (2018). Enforceable and unenforceable laws in agribusiness systems. *RAUSP Management Journal*, *53*(2), 178–189. https://doi.org/10.1016/j.rauspm.2017.06.002

Backus, J. W., Bauer, F. L., Green, J., Naur, P., Rutishauser, H., Mccarthy, J., Perlis, A. J., Vauquois, B., Wegstein, J. H., Wijngaarden, A. V., & Woodger, M. (1960). Report on the Algorithmic Language ALGOL 60. *Communications of the ACM*, *3*(5), 16.

Bainbridge, L. (1982). Ironies of Automation. *IFAC Proceedings Volumes*, *15*(6), 129–135. https://doi.org/10.1016/S1474-6670(17)62897-0

Baker, C., Nancarrow, C., & Tinson, J. (2005). The Mind versus Market Share Guide. *International Journal of Market Research*, *47*(5), 523–540. https://doi.org/10.1177/147078530504700505

Baker, D. R. (2004). How to Use a Decision Table Methodology for the Analysis of Complex Conditional Actions Requirements. *Methods & Tools*, *12*(3), 23–35.

Baker, G. P., & Hacker, P. M. S. (2009). *Wittgenstein: Rules, Grammar and Necessity: Volume 2 of an Analytical Commentary on the Philosophical Investigations, Essays and Exegesis §§185-242*. John Wiley & Sons.

Bargh, J., & Chartrand, T. (1999). The unbearable automaticity of being. *American Psychologist*, *54*(7), 462–479. https://doi.org/10.1037/0003-066X.54.7.462

Barrouillet, P., & Fayol, M. (1998a). From algorithmic computing to direct retrieval: Evidence from number and alphabetic arithmetic in children and adults. *Memory & Cognition*, *26*(2), 355–368. https://doi.org/10.3758/BF03201146

Barrouillet, P., & Fayol, M. (1998b). From algorithmic computing to direct retrieval: Evidence from number and alphabetic arithmetic in children and adults. *Memory & Cognition*, *26*(2), 355–368. https://doi.org/10.3758/BF03201146

Bartlett, F. (1932a). *Remembering: A Study in Experimental and Social Psychology*. Cambridge University Press.

Bartlett, F. (1932b). *Remembering: A Study in Experimental and Social Psychology (Republished in 1995)* (2 edition). Cambridge University Press.

Bartlett, J. (2018, August 15). *How AI could kill off democracy*. https://www.newstatesman.com/science-tech/technology/2018/08/how-ai-could-kill-democracy-0

Batanero, C., & Díaz, C. (2006). *Methodological and didactical controversies around statistical inference. Actes du 36iémes Journées de la Societé Française de Statistique*. Actes du 36iémes Journées de la Societé Française de Statistique. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9c5cea1a5b2c1a94692215589185a88149188505

Bazzan, G. (2021). *Effective Governance Designs of Food Safety Regulation in the EU: Do Rules Make the Difference?* Springer Nature.

Beer, S. (1981). *Brain of the Firm*. John Wiley & Sons.

Belnap, N. D. (1959). *A Formalization of Entailment* [Ph.D.]. Yale Unversity.

Belnap, N. D. (1977a). A Useful Four-Valued Logic. In J. M. Dunn & G. Epstein (Eds.), *Modern Uses of Multiple-Valued Logic* (pp. 5–37). Springer Netherlands.

https://doi.org/10.1007/978-94-010-1161-7_2

Belnap, N. D. (1977b). How a Computer Should Think. In H. Omori & H. Wansing (Eds.), *New Essays on Belnap-Dunn Logic. 2020.* (pp. 35–53). Springer International Publishing. https://doi.org/10.1007/978-3-030-31136-0_4

Belshaw, D. (1981). A Theoretical Framework For Data-Economising Appraisal Procedures, with Applications to Rural Development Planning. *IDS Bulletin. Institute for Development Studies*, *12*, 12–22. https://doi.org/10.1111/j.1759-5436.1981.mp12004004.x

Benenson, Y., Adar, R., Paz-Elizur, T., Livneh, Z., & Shapiro, E. (2003). DNA molecule provides a computing machine with both data and fuel. *Proceedings of the National Academy of Sciences*, *100*(5), 2191–2196. https://doi.org/10.1073/pnas.0535624100

Benet, J. (2014). *IPFS - Content Addressed, Versioned, P2P File System. Version 3.0*. https://arxiv.org/abs/1407.3561

Benet, J. (2021). *Content Addressing and CIDs on the Interplanetary File System (IPFS)*. https://docs.ipfs.io/concepts/content-addressing/

Bennett, C. H. (1982). The thermodynamics of computation—A review. *International Journal of Theoretical Physics*, *21*(12), 905–940. https://doi.org/10.1007/BF02084158

Berners-Lee, T. (1998a). *Principles of Design (Last edited 2013)*. https://www.w3.org/DesignIssues/Principles.html

Berners-Lee, T. (1998b). *The World Wide Web and the "Web of Life."* https://www.w3.org/People/Berners-Lee/UU.html

Berners-Lee, T., & Connolly, D. (2008). *Notation3 (N3): A readable RDF syntax*. https://www.w3.org/TeamSubmission/n3/

Berners-Lee, T., Fielding, R., & Masinter, L. (2005). *RFC 3986: Uniform Resource Identifier (URI): Generic Syntax*. https://datatracker.ietf.org/doc/html/rfc3986/

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*(5), 34–43.

Berners-Lee, T., & Mendelsohn, N. (2006, February 23). *The Rule of Least Power*. https://www.w3.org/2001/tag/doc/leastPower.html

Bernus, P., Goranson, T., Gøtze, J., Jensen-Waud, A., Kandjani, H., Molina, A., Noran, O., Rabelo, R. J., Romero, D., Saha, P., & Turner, P. ((In Press)). Enterprise engineering and management at the crossroads. *Computers in Industry*. https://doi.org/10.1016/j.compind.2015.07.010

Berry, D. (2018, April 27). *Infrasomatization and the Exospherical Technical System*. http://zkm.de/en/media/video/encoding-cultures-david-m-berry-automating-thought-infrasomatization-and-the-exospherical-technical

Berry, D. (2019). Against infrasomatization: Towards a critical theory of algorithms. In D. Bigo, E. Isin, & R. Evelyn (Eds.), *Data politics: Worlds, subjects, rights*. Routledge.

Bhamra, R., Dani, S., & Burnard, K. (2011). Resilience: The concept, a literature review and future directions. *International Journal of Production Research*, *49*(18), 5375–5393. https://doi.org/10.1080/00207543.2011.563826

Bidgoli, H. (2015). *Management Information Systems: MIS* (5th ed.). Cengage Learning.

Bies, R. J., Bartunek, J. M., Fort, T. L., & Zald, M. N. (2007). Corporations as social change agents: Individual, interpersonal, institutional, and environmental dynamics. *Academy of*

Management Review, *32*(3), 788–793. https://doi.org/10.5465/amr.2007.25275515

Bimbó, K., & Dunn, J. M. (2001). Four-valued Logic. *Notre Dame Journal of Formal Logic*, *42*(3). https://doi.org/10.1305/ndjfl/1063372199

BioBricks Foundation. (2021). *The BioBrick™ Public Agreement (BPA)*. https://biobricks.org/bpa/

BIS. (2008). Highlights of international banking and financial market activity. *BIS Quarterly Review, December*. http://www.bis.org/publ/qtrpdf/r_qt0806b.pdf

Bloch, P. H. (1995). Seeking the Ideal Form: Product Design and Consumer Response. *Journal of Marketing*, *59*(3), 16–29.

Block, C. (2022, March 16). 12 Reasons Your Digital Transformation Will Fail. *Forbes Coaches Council*. https://www.forbes.com/sites/forbescoachescouncil/2022/03/16/12-reasons-your-digital-transformation-will-fail/

Bloor, D. (1997). *Wittgenstein, Rules and Institutions*. Routledge, London. https://www.routledge.com/Wittgenstein-Rules-and-Institutions/Bloor/p/book/9780415161480

Bobulescu, R. (2015). From Lotka's biophysics to Georgescu-Roegen's bioeconomics. *Ecological Economics*, *120*, 194–202. https://doi.org/10.1016/j.ecolecon.2015.10.016

Boley, H. (2006). The RuleML Family of Web Rule Languages. In J. J. Alferes, J. Bailey, W. May, & U. Schwertel (Eds.), *Principles and Practice of Semantic Web Reasoning* (Vol. 4187, pp. 1–17). Springer Berlin Heidelberg. https://doi.org/10.1007/11853107_1

Boley, H., Paschke, A., Athan, T., Guirca, A., Bassiliades, N., Guido Governatori, Palmirani, M., Wyner, A., Kozlenkov, A., & Zou, G. (2017). *Specification of RuleML 1.03*. RuleML Wiki. http://wiki.ruleml.org/index.php/Specification_of_RuleML_1.03

Boley, H., Paschke, A., & Shafiq, O. (2010). RuleML 1.0: The Overarching Specification of Web Rules. In M. Dean, J. Hall, A. Rotolo, & S. Tabet (Eds.), *Semantic Web Rules* (Vol. 6403, pp. 162–178). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-16289-3_15

Bormann, C., & Hoffman, P. (2013). *RFC 7049. Concise Binary Object Representation (CBOR)*. https://tools.ietf.org/html/rfc7049

Bosak, J. (1996). *XML at SGML 96*. https://lists.w3.org/Archives/Public/w3c-sgml-wg/1996Nov/0241.html

Boyd, A. B. (2018). *Thermodynamics of Correlations and Structure in Information Engines* [PhD Thesis, University of California, Davis]. https://www.proquest.com/docview/2047668591

Boys, V. (1956). The Soap Bubble. In J. Newman (Ed.), *The World of Mathematics* (Vol. 2, pp. 891–900). Simon & Schuster.

Bradner, S. (1997). *BCP14. Key words for use in RFCs to Indicate Requirement Levels. Internet Engineering Task Force (IETF)*. https://tools.ietf.org/html/bcp14

Brandenburger, A., & Nalebuff, B. (1997). *Co-Opetition* (1 edition). Crown Business.

Brandom, R. (2008). *Between saying and doing: Towards an analytic pragmatism*. Oxford University Press.

Bratman, M. (2014). *Shared Agency: A Planning Theory of Acting Together*. Oxford University Press.

Bray, T. (2014). *RFC 7159. The JavaScript Object Notation (JSON) Data Interchange Format*. Internet Engineering Task Force. https://tools.ietf.org/html/rfc7159

Bredillet, C. (2005). Understanding the very nature of project management: A praxiological approach. *Innovations: Project Management Research*. http://eprints.qut.edu.au/49501/

Breidenbach, L., Cachin, C., Coventry, A., Ellis, S., Juels, A., Miller, A., Magauran, B., Nazarov, S., Topliceanu, A., Zhang, F., Chan, B., Koushanfar, F., Moroz, D., & Tramer, F. (2021). *Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks*. Chainlink. https://research.chain.link/whitepaper-v2.pdf?_ga=2.167498878.1551014398.1648834107-903530017.1648834107

Breiman, L. (2001). Statistical Modeling: The Two Cultures. *Statistical Science*, *16*(3), 199–231.

Brend, Y. (2020, July 3). Cross-border mingling near Peace Arch still a walk in the park, says lawyer from Washington state. In *CBC*. https://www.cbc.ca/news/canada/british-columbia/peace-arch-us-side-picnic-access-for-canadians-1.5635031

Breslin, J., Romano, D., & Percival, J. (2015). Conceptualizing and Modeling Multi-Level Organizational Co-Evolution. In D. Secchi & M. Neumann (Eds.), *Agent-Based Simulation of Organizational Behavior, New Frontiers of Social Science Research*. Springer.

Brewer, C. (2013). *ColorBrewer: Color Advice for Maps*. https://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3

Brodie, L. (2004). *Thinking Forth: A Language and Philosophy for Solving Problems, 2nd Edition*. http://thinking-forth.sourceforge.net/

Brooks, F. (1995). *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition*. Pearson Education.

Brown, H. (2015). *The Father of PLC's: How Richard Morley Revolutionised the Automation Industry*. Radwell International. https://blog.radwell.com/2015/11/25/the-father-of-plcshow-richard-morley-revolutionised-the-automation-industry

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. In H. Larochelle (Ed.), *Advances in Neural Information Processing Systems*. arXiv. http://arxiv.org/abs/2005.14165

Browning, L. (2013, June 19). *The biggest winners of an Internet sales tax*. Fortune. http://fortune.com/2013/06/19/the-biggest-winners-of-an-internet-sales-tax/

Brownlee, K. (2012). *Conscience and Conviction: The Case for Civil Disobedience*. OUP Oxford.

Bry, F., & Marchiori, M. (2005). Ten Theses on Logic Languages for the Semantic Web FagesSylvain Soliman. In F. Fages & S. Soliman (Eds.), *Principles and Practice of Semantic Web Reasoning* (Vol. 3703, pp. 42–49). Springer Berlin Heidelberg. https://doi.org/10.1007/11552222_5

Bulman-Pozen, J., & Poz, D. (2015). Uncivil Obedience. *Columbia Law Review*, *115*(4), 809–872.

Bush, R., & Meyer, D. (2002). *RFC 3439. Some Internet Architectural Guidelines and Philosophy*. Internet Engineering Task Force. https://tools.ietf.org/html/rfc3439#page-3

California Court of Appeal. (2022). *Almond Alliance of California v. Fish & Game Commission et al. (Case No. C093542)*. 3rd Appellate District. https://www.courts.ca.gov/opinions/documents/C093542.PDF

Çalışkan, K. (2009). The Meaning of Price in World Markets. *Journal of Cultural Economy*, *2*(3), 239–268. https://doi.org/10.1080/17530350903345462

Campbell-Kelly, M., Croarken, M., Flood, M., & Robson, E. (Eds.). (2004). *The History of Mathematical Tables: From Sumer to Spreadsheets*. Oxford University Press. https://oxford.universitypressscholarship.com/view/10.1093/acprof:oso/9780198508410.001.0001/acprof-9780198508410

Cao, B., Yin, J., Zhang, Q., & Ye, Y. (2010). A MapReduce-Based Architecture for Rule Matching in Production System. *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, 790–795. https://doi.org/10.1109/CloudCom.2010.11

Cao, D. (1999). “Ought to” as a Chinese Legal Performative? *International Journal for the Semiotics of Law*, *12*(2), 151–167.

Carlin, B. I. (2009). Strategic price complexity in retail financial markets. *Journal of Financial Economics*, *91*(3), 278–287. https://doi.org/10.1016/j.jfineco.2008.05.002

Carpenter, B. (1996). *RFC 1958. Architectural Principles of the Internet*. Internet Architecture Board. https://www.ietf.org/rfc/rfc1958.txt

Carpenter, J., & Hewitt, E. (2016). *Cassandra: The Definitive Guide* (2nd ed.). O’Reilly Media, Inc.

Carroll, J., Copestake, A., Malouf, R., & Oepen, S. (2020). *Linguistic Knowledge Builder (LKB). Online application. John Carroll, Ann Copestake, Robert Malouf, Stephan Oepen*. http://moin.delph-in.net/wiki/LkbTop

Cartwright, N. (2020). Middle-range theory: Without it what could anyone do? *THEORIA. An International Journal for Theory, History and Foundations of Science*, *35*(3), 269–323. https://doi.org/10.1387/theoria.21479

Castañeda, H.-N. (1989). Paradoxes of moral reparation: Deontic foci vs. circumstances. *Philosophical Studies*, *57*(1), 1–21. https://doi.org/10.1007/BF00355659

Çelik, T., Lilley, C., & Baron, L. D. (2018, June 19). *CSS Color Module Level 3. W3C Recommendation* [Text]. W3C. https://www.w3.org/TR/2018/REC-css-color-3-20180619/

Chakrabarti, A., & Lindemann, U. (2015). *Impact of Design Research on Industrial Practice: Tools, Technology, and Training*. Springer.

Chambers, J. M. (1993). Greater or lesser statistics: A choice for future research. *Statistics and Computing*, *3*(4), 182–184. https://doi.org/10.1007/BF00141776

Chambers, J. M. (1998). *Programming with Data: A Guide to the S Language*. Springer Science & Business Media.

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., & Gruber, R. E. (2008). Bigtable: A Distributed Storage System for Structured Data. *ACM Transactions on Computer Systems*, *26*(2), 1–26. https://doi.org/10.1145/1365815.1365816

Chapin, D. (2008, March). *SBVR: What is Now Possible and Why?* Business Rules Journal. Vol. 9, No. 3. An Online Publication of the Business Rules Community . https://www.brcommunity.com/articles.php?id=b407

Chatila, R., & Havens, J. C. (2019). The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. In M. I. Aldinhas Ferreira, J. Silva Sequeira, G. Singh Virk, M. O. Tokhi, & E. E. Kadar (Eds.), *Robotics and Well-Being* (Vol. 95, pp. 11–16). Springer International Publishing. https://doi.org/10.1007/978-3-030-12524-0_2

Chen, C., Wu, Q., Ke, Q., Wang, T., Zhang, Y., Wei, F., Wang, X., & Liu, G. (2022). Implementation of novel boolean logic gates for IMPLICATION and XOR functions using riboregulators. *Bioengineered*, *13*(1), 1235–1248. https://doi.org/10.1080/21655979.2021.2020493

Chen, C., Xiao, W., Zhao, J., Zhang, Z., & Shi, X. (2020). DNA Logic Circuits Based on Accurate Step Function Gate. *IEEE Access*, *8*, 125513–125520. https://doi.org/10.1109/ACCESS.2020.3003636

Chile, B. B. del C. N. de. (2021). *Comparador de Constituciones. Proceso Constituyente | Biblioteca del Congreso Nacional de Chile* [Text]. bcn.cl; BCN. Biblioteca del Congreso Nacional de Chile. http://www.bcn.cl/procesoconstituyente/comparadordeconstituciones/home

Chomsky, N. (1955). *Transformational Analysis* [PhD (Linguistics), MIT]. https://repository.upenn.edu/dissertations/AAI0013380/

Chomsky, N. (1956). *The Logical Structure of Linguistic Theory (published in book form in 1975)*. MIT and Plenum Press.

Chomsky, N. (1957). *Syntactic Structures (2nd Edition, 2000)*. Mouton de Gruyter.

Chomsky, N. (1958). Finite State Languages. *Information and Control*, *1*, 91–112.

Chomsky, N. (2000). *Language and Mind (3rd Ed.)*. Cambridge University Press.

Church, A. (1956). *Introduction to mathematical logic 1. 1*. Princeton Univ. Press.

Clark, D. (1988). The Design Philosophy of the DARPA Internet Protocols. Proc. SIGCOMM '88. *Computer Communication Review*, *18*(4), 106–114.

Clark, D. (2018). *Designing an Internet*. The MIT Press. https://mitpress.mit.edu/books/designing-internet

Cleveland, W. S. (2014). Data science: An action plan for expanding the technical areas of the field of statistics: Technical Areas of the Field of Statistics. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *7*(6), 414–417. https://doi.org/10.1002/sam.11239

Cloudflare. (2021). *IPFS Gateway · Cloudflare Distributed Web Gateway*. https://developers.cloudflare.com/distributed-web/ipfs-gateway

Coeckelbergh, M., & Funk, M. (2018). Wittgenstein as a Philosopher of Technology: Tool Use, Forms of Life, Technique, and a Transcendental Argument. *Human Studies*, *41*(2), 165–191. https://doi.org/10.1007/s10746-017-9452-6

Coenen, F. (1999). *Declarative Languages* [University of Liverpool]. Course Sheet 2CS24. http://cgi.csc.liv.ac.uk/~frans/OldLectures/2CS24/declarative.html#detail

Coleman, V. (2016). Embodied Molecular Computation: Potential and Challenges. *Computer*, *49*(9), 43–51. https://doi.org/10.1109/MC.2016.295

Connelly, J. (2008). *CPE Report—Ternary Computing Testbed: 3!Trit Computer Architecture*. Computer Engineering Department , California Polytechnic State University of San Luis Obispo. http://xyzzy.freeshell.org/trinary/CPE%20Report%20-%20Ternary%20Computing%20Testbed%20-%20RC6a.pdf

Conrad, D. A. (1988). *Rule-Based Explanations and the Philosophy of Mind* [PhD Thesis, Wayne State University]. https://philpapers.org/rec/CONREA-3

Conrad, M. (1986). The lure of molecular computing: While marketable products seem decades away, researchers are crystallizing theories and devices that will give biological organisms the power to compute. *IEEE Spectrum*, *23*(10), 55–60. https://doi.org/10.1109/MSPEC.1986.6371116

Cook, W. W. (1924). The Logical and Legal Bases of the Conflict of Laws. *The Yale Law Journal*, *33*(5), 457. https://doi.org/10.2307/788019

Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. CSLI Publications.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal Recursion Semantics: An Introduction. *Research on Language and Computation*, *3*(2–3), 281–332. https://doi.org/10.1007/s11168-006-6327-9

Copola Azenha, F., Aparecida Reis, D., & Leme Fleury, A. (2021). The Role and Characteristics of Hybrid Approaches to Project Management in the Development of Technology-Based Products and Services. *Project Management Journal*, *52*(1), 90–110. https://doi.org/10.1177/8756972820956884

Cosulschi, M. (2015). *Rulestore API*. RuleML. http://wiki.ruleml.org/index.php/Rulestore_API

Couto, M., Pereira, R., Ribeiro, F., Rua, R., & Saraiva, J. (2017). *Towards a Green Ranking for Programming Languages*. 8.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, *24*(1), 87–114. https://doi.org/10.1017/S0140525X01003922

Crabtree, G. (2013a). Our fragile intellect. Part I. *Trends in Genetics*, *29*(1), 1–3. https://doi.org/10.1016/j.tig.2012.10.002

Crabtree, G. (2013b). Our fragile intellect. Part II. *Trends in Genetics*, *29*(1), 3–5. https://doi.org/10.1016/j.tig.2012.10.003

Crabtree, G. (2013c). Our fragile intellect: Response to Dr Mitchell. *Trends in Genetics*, *29*(2), 60–62. https://doi.org/10.1016/j.tig.2012.12.005

Crabtree, G. (2013d). Rethinking our intellectual origins: Response to Kalinka et al. *Trends in Genetics*, *29*(3), 127–129. https://doi.org/10.1016/j.tig.2013.01.013

Crawford, L., & Duncan, B. (Eds.). (2007). *A Framework for Performance Based Competency Standards for Global Level 1 and 2 Project Managers, Version 1.7a*. Global Alliance for Project Performance Standards.

Crawford, S. E. S., & Ostrom, E. (1995). A Grammar of Institutions. *American Political Science Review*, *89*(3), 582–600. https://doi.org/10.2307/2082975

Crossley, J. N., & Henry, S. (1990). Thus Spake al-Khwsrizmi: A Translation of the Text of Cambridge University Library Ms. Ii.vi.5. *Historia Mathematica*, *17*, 103–131.

Cunneyworth, W. (1994). *Table Driven Design: A development strategy for minimal maintenance information systems*. DataKinetics Inc. http://www.dkl.com/wp-content/uploads/2016/05/DataKinetics-Table-Driven-Design.pdf

Cunneyworth, W., Kelly, D., & Kim, T. (2022, June 29). *Replies to a Request for Comment on the "Data with Direction" Thesis by Joseph Potvin, Doctoral Candidate, UQO*. https://gitlab.com/xalgorithms-alliance/data-with-direction-specification/dwds-documents/-/issues

Cunneyworth, W., & Olders, W. (2020, June 4). *Searching Trinary Decision Tables (personal communication)* [Personal communication].

Dalchau, N., Szép, G., Hernansaiz-Ballesteros, R., Barnes, C. P., Cardelli, L., Phillips, A., & Csikász-Nagy, A. (2018). Computing with biological switches and clocks. *Natural Computing*, *17*(4), 761–779. https://doi.org/10.1007/s11047-018-9686-x

Damjanović, Z. M. (1998). Logic core of genetic code. *Glasnik of the Sec. of Nat. Sci. of Montenegrin Academy of Sciences and Art (CANU)*, *12*, 5–8.

Damjanović, Z. M., & Rakočević, M. M. (2005). Genetic Code: An Alternative Model of Translation. *Annals of the New York Academy of Sciences*, *1048*(1), 517–523. https://doi.org/10.1196/annals.1342.081

Damonte, A., & Bazzan, G. (2022, April 19). *Rules as Data Workshop*. ECPR Joint Sessions, University of Edinburgh. https://ecpr.eu/Events/Event/PanelDetails/11396

Das, R., Hanson, J. E., Kephart, J. O., & Tesauro, G. (2001). Agent-Human Interactions in the Continuous Double Auction. *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, 8. http://spider.sci.brooklyn.cuny.edu/~parsons/courses/840-spring-2005/notes/das.pdf

Davis, M., & Davis, V. (2005). The Jacquard and the Computer. *The Journal of Cloth and Culture*, *3*(1), 76–87.

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters (Originally published in 2004 as a technical white paper by Google Inc.). *Communications of the ACM*, *51*(1), 107–113. https://doi.org/10.1145/1327452.1327492

Dean, T. J., & McMullen, J. S. (2002). Market failure and entrepreneurlial opportunity. *Academy of Management Proceedings*, *2002*(1), F1–F6. https://doi.org/10.5465/APBPP.2002.7516617

Delve, J. (2007). Joseph Marie Jacquard: Inventor of the Jacquard Loom. *IEEE Annals of the History of Computing*, *29*(99), 98–102. https://doi.org/10.1109/MAHC.2007.4343546

DEON. (2019). *15th International Conference on Deontic Logic and Normative Systems (DEON 2020), 30 July – 2 August 2020. Hosted by the Munich Center for Mathematical Philosophy (MCMP), Ludwig-Maximilians-Universität München.* https://www.mcmp.philosophie.uni-muenchen.de/events/workshops/container/deon-2020/index.html

Deryck, M., Aerts, B., & Vennekens, J. (2019). Adding Constraint Tables to the DMN Standard: Preliminary Results. In P. Fodor, M. Montali, D. Calvanese, & D. Roman (Eds.), *Rules and Reasoning* (Vol. 11784, pp. 171–179). Springer International Publishing. https://doi.org/10.1007/978-3-030-31095-0_12

Dias de Figueiredo, A. (2018, March). *Action Research and Design Research March 2018 DOI: 10.13140/RG.2.2.34006.19526* . PhD Course on Qualitative Research (Philosophical Foundations of Engineering and Technology), University of Coimbra, Portugal. https://www.researchgate.net/publication/337926074_Action_Research_and_Design_Research

Diebold, F. X. (2012). On the Origin(s) and Development of the Term “Big Data.” *SSRN Electronic Journal*, *PIER Working Paper No. 12-037*((The latest version of this paper is maintained at https://arxiv.org/pdf/2008.05835.pdf)). https://doi.org/10.2139/ssrn.2152421

Dillard, J. F., & Ruchala, L. (2005). The rules are no game: From instrumental rationality to administrative evil. *Accounting, Auditing & Accountability Journal*, *18*(5), 608–630. https://doi.org/10.1108/09513570510620475

Dille, T., & Söderlund, J. (2011). Managing inter-institutional projects: The significance of isochronism, timing norms and temporal misfits. *International Journal of Project Management*, *29*(4), 480–490. https://doi.org/10.1016/j.ijproman.2011.02.007

Dinishak, J. (2014). ‘Blind’ to the obvious: Wittgenstein and Köhler on the obvious and the hidden. *History of the Human Sciences*, *27*(4), 59–76. https://doi.org/10.1177/0952695114530393

Dong, T., Cheng, Q., Cao, B., & Shi, J. (2018). A Novel Approach to Distributed Rule Matching and Multiple Firing Based on MapReduce: *Journal of Database Management*, *29*(2), 62–84.

https://doi.org/10.4018/JDM.2018040104

Donoho, D. (2017). 50 Years of Data Science. *Journal of Computational and Graphical Statistics*, *26*(4), 745–766. https://doi.org/10.1080/10618600.2017.1384734

Dopfer, K. (2012). The origins of meso economics: Schumpeter's legacy and beyond. *Journal of Evolutionary Economics*, *22*(1), 133–160. https://doi.org/10.1007/s00191-011-0218-4

Dopfer, K., Foster, J., & Potts, J. (2004). Micro-meso-macro. *Journal of Evolutionary Economics*, *14*(3), 263–279. https://doi.org/10.1007/s00191-004-0193-0

Driouchi, T., & Bennett, D. (2012). Real Options in Management and Organizational Strategy: A Review of Decision-making and Performance Implications: Management and Organizational Strategy. *International Journal of Management Reviews*, *14*(1), 39–62. https://doi.org/10.1111/j.1468-2370.2011.00304.x

Dubois, D., Hájek, P., & Prade, H. (2000). Knowledge-Driven versus Data-Driven Logics. *Journal of Logic, Language, and Information*, *9*, 65–89.

Dunn, J. M. (1966). *The Algebra of Intensional Logic* [Ph.D.]. University. of Pittsburgh.

Dunn, J. M. (2019). Two, Three, Four, Infinity: The Path to the Four-Valued Logic and Beyond. In H. Omori & H. Wansing (Eds.), *New Essays on Belnap-Dunn Logic* (Vol. 418, pp. 77–97). Springer International Publishing. https://doi.org/10.1007/978-3-030-31136-0_6

Eastman, W., & Bailey, J. R. (1998). Mediating the Fact-Value Antinomy: Pattems in Managerial and Legal Rhetoric, 1890-1990. *Organization Science*, *9*(2), 232–245.

Ebrahimi, S. A., Reshadinezhad, M. R., Bohlooli, A., & Shahsavari, M. (2016). Efficient CNTFET-based design of quaternary logic gates and arithmetic circuits. *Microelectronics Journal*, *53*, 156–166. https://doi.org/10.1016/j.mejo.2016.04.016

Ehrgott, M. (2012). Vilfredo Pareto and multi-objective optimization. In M Grötschel (ed.), Optimization stories: 21st International Symposium on Mathematical Programming, Berlin, August 19—24, 2012. *Documenta Mathematica - Deutschen Mathematiker-Vereiningung, Extra , Bielefeld.*, *21*, 447–453.

Eigen, Z. J., Sherwyn, D. S., & Menillo, N. F. (2015). When Rules are Made to Be Broken. *Northwestern University Law Review*, *109*(1), 109–171.

Ellis, S., Juels, A., & Nazarov, S. (2017). *Chainlink_Ellis-Juels-Nazarov-2017_whitepaper-v1.pdfChainLink A Decentralized Oracle Network Steve Ellis, Ari Juels † , and Sergey Nazarov 4 September 2017*. Chainlink. https://research.chain.link/whitepaper-v1.pdf?_ga=2.267517390.1551014398.1648834107-903530017.1648834107

Elqayam, S., & Evans, J. St. B. T. (2011). Subtracting "ought" from "is": Descriptivism versus normativism in the study of human thinking. *Behavioral and Brain Sciences*, *34*(05), 233–248. https://doi.org/10.1017/S0140525X1100001X

Emirbayer, M., & Mische, A. (1998). What Is Agency? *American Journal of Sociology*, *103*(4), 962–1023.

Endo, K. (1994). The Principle of Subsidiarity: From Johannes Althusius to Jacques Delors. **√ 이측정된상**, *44*(6), 652–553.

Enfield, N., & Kockelman, P. (Eds.). (2017). *Distributed Agency*. Oxford University Press.

Engeström, Y. (2009). The Future of Activity Theory: A Rough Draft. In A. Sannino, H. Daniels, & K. D. Gutiérrez (Eds.), *Learning and Expanding with Activity Theory* (1st ed., pp. 303–328). Cambridge University Press. https://doi.org/10.1017/CBO9780511809989.020

Engeström, Y., & Glăveanu, V. (2012). On Third Generation Activity Theory: Interview With Yrjö Engeström. *Europe's Journal of Psychology*, *8*(4), 515–518. https://doi.org/10.5964/ejop.v8i4.555

Enstrom, D. W. (2016). *A Simplified Approach to It Architecture with BPMN: A Coherent Methodology for Modeling Every Level of the Enterprise*. iUniverse.

Epstein, J. M. (2008, October 31). *Why Model?* [Text.Article]. JASSS. https://www.jasss.org/11/4/12.html

Erbas-Cakmak, S., Kolemen, S., Sedgwick, A. C., Gunnlaugsson, T., James, T. D., Yoon, J., & Akkaya, E. U. (2018). Molecular logic gates: The past, present and future. *Chemical Society Reviews*, *47*(7), 2228–2248. https://doi.org/10.1039/C7CS00491E

Eriksson, K.-E., Lindgren, K., & Månsson, B. (1987). *Structure, Context, Complexity, Organization: Physical Aspects of Information and Value*. World Scientific. http://kau.diva-portal.org/smash/record.jsf?pid=diva2:597625

Espedal, B. (2007). Why rules rather than discretion: When the leadership intends to transform a desired policy into reality. *Journal of Organizational Change Management*, *20*(1), 95–108. https://doi.org/10.1108/09534810710715306

Ethiopia. (2019). *Aircraft Accident Investigation Bureau Preliminary Report No. AI-01/19. RE: Ethiopian Airlines flight 302, Boeing 737-8 (MAX) on March 10, 2019, at 05:38 UTC.* Ethiopia, Ministry of Transport. http://www.ecaa.gov.et/documents/20435/0/Preliminary+Report+B737-800MAX+,(ET-AVJ).pdf

Euler, L. (1768). *Lettres à une Princesse d'Allemagne* (Vol. 2). Imprimerie de Académie Impériale des Sciences.

Evanko, D. (2014, February 27). Guidelines for algorithms and software. *Nature Methods: Methagora*. http://blogs.nature.com/methagora/2014/02/guidelines-for-algorithms-and-software-in-nature-methods.html

Evans, J. S. B. T. (1984). Heuristic and analytic processes in reasoning. *British Journal of Psychology*, *75*(4), 451–468.

Fahlman, S. E. (1977). *A System for Representing and Using Real-World Knowledge*. http://dspace.mit.edu/handle/1721.1/6888

Fan, X. (2015). *Real-time embedded systems: Design principles and engineering practices*. Elsevier.

Farsad, N., Yilmaz, H. B., Eckford, A., Chae, C.-B., & Guo, W. (2016). A Comprehensive Survey of Recent Advancements in Molecular Communication. *IEEE Communications Surveys & Tutorials*, *18*(3), 1887–1919. https://doi.org/10.1109/COMST.2016.2527741

Faste, T., & Faste, H. (2012). Demystifying "design research": Design is not research, research is design. *Proceedings of the IDSA*. http://www.academia.edu/download/30962088/demystifying_design_research.pdf

Faster Payments Task Force. (2017). *The U.S. Path to Faster Payments. Final Report Part Two: A Call to Action.* US Federal Reserve.

Fayezi, S., O'Loughlin, A., & Zutshi, A. (2012). Agency Theory and Supply Chain Management: A Structured Literature Review. *Supply Chain Management: An International Journal*, *17*(5), 556–570.

Fehrer, J. A., Woratschek, H., & Brodie, R. J. (2018). A systemic logic for platform business models. *Journal of Service Management*, *29*(4), 546–568. https://doi.org/10.1108/JOSM-02-2017-0036

Feyerabend, P. (1982). *Science in a Free Society*. Verso.

Feyerabend, P. (1993). *Against Method*. Verso.

Feyerabend, P. K. (1970). Consolations for the Specialist. In A. Musgrave & I. Lakatos (Eds.), *Criticism and the Growth of Knowledge. Proceedings of the International Colloquium in the Philosophy of Science* (Vol. 4, pp. 197–230). Cambridge University Press.

Feyerabend, P. K. (2011). *The Tyranny of Science*. Wiley.

Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics*, *21*(6/7), 22.

Feynman, R. P. (2005). The Computing Machines in the Future. Nishina Memorial Lecture at Gakushuin University (Tokyo), on August 9, 1985. Nishina Foundation. In *The Pleasure of Finding Things Out: The Best Short Works of Richard P. Feynman*. Basic Books.

FIDO Alliance. (2020). *Interoperability Testing*. FIDO Alliance. https://fidoalliance.org/certification/interoperability-testing/

Fielding, J. (2013). *In Search of Ineffable: Wittgenstein on Time, Change and History (PhD Dissertation)* [Université Paris I, Panthéon-Sorbonne, France]. http://www.jamesmfielding.com/research.html

Filatova, M. V., Stukalo, O. G., Lebedeva, L. V., Tsukanova, K. A., & Dzakhmisheva, I. S. (2021). Project management in the digital economy. *Proceedings of the Voronezh State University of Engineering Technologies*, *82*(4), 335–339. https://doi.org/10.20914/2310-1202-2020-4-335-339

Finlayson, S. G., Bowers, J. D., Ito, J., Zittrain, J. L., Beam, A. L., & Kohane, I. S. (2019). Adversarial attacks on medical machine learning. *Science*, *363*(6433), 1287–1289. https://doi.org/10.1126/science.aaw4399

Fitting, M. (1988). First Order Modal Tableaux. *Journal of Automated Reasoning*, *4*, 191–213.

Fletcher, G. P. (1985). Paradoxes in Legal Thought. *Columbia Law Review*, *85*(6), 1263. https://doi.org/10.2307/1122394

Floricel, S., Bonneau, C., Aubry, M., & Sergi, V. (2014). Extending project management research: Insights from social theories. *International Journal of Project Management*, *32*(7), 1091–1107. https://doi.org/10.1016/j.ijproman.2014.02.008

Floyd, M. (2009, August 15). *Web Techniques: A Conversation with Charles F. Goldfarb*. Web Techniques Magazine (via Wayback Machine). https://web.archive.org/web/20090815103836/http://www.beyond-html.com/columns/1198/1198.html

Foropon, C., & McLachlin, R. (2013). Metaphors in operations management theory building. *International Journal of Operations & Production Management*, *33*(2), 181–196. https://doi.org/10.1108/01443571311295626

Forrester, J. (1961). *Industrial Dynamics* (1st ed.). Martino Fine Books.

Fowler, M. (2009, January 7). *Should I use a Rules Engine?* Martinfowler.Com. https://martinfowler.com/bliki/RulesEngine.html

Fowler, M. (2013, August 21). *GivenWhenThen*. Martinfowler.Com. https://martinfowler.com/bliki/GivenWhenThen.html

Frantz, C. (2023, January 8). *Direct email correspondence about approaches to a typology of controlled natural language frameworks* [Personal communication].

Frantz, C. K., & Siddiki, S. (2022). *Institutional Grammar: Foundations and Applications for Institutional Analysis*. Springer International Publishing. https://doi.org/10.1007/978-3-030-86372-2

Frantz, C., & Siddiki, S. (2022). *Understanding Regulatory Rules using the Institutional Grammar 2.0* (A. Damonte & G. Bazzan, Eds.). Wiley (Special Issue of 'Regulation and Governance'). https://ecpr.eu/Events/Event/PanelDetails/11396

Free Software Foundation, (first). (1996). *What is free software?* http://www.gnu.org/philosophy/free-sw.html

Frege, G. (1879). Begriffsschrift (translated by M. Beaney). In M. Beaney (Ed.), *The Frege Reader (1997)* (pp. 47–78). Blackwell Publishers.

Frege, G. (1892). On Sense and Reference (Über Sinn und Bedeutung) (M. Black, Trans.). *Zeitschrift Für Philosophie Und Philosophische Kritik*, *100*, 25–50.

Frey, C. B., & Osborne, M. A. (2017). The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change*, *114*, 254–280. https://doi.org/10.1016/j.techfore.2016.08.019

Fu, W., Wang, Q., & Zhao, X. (2018a). Platform-based service innovation and system design: A literature review. *Industrial Management & Data Systems*, *118*(5), 946–974. https://doi.org/10.1108/IMDS-03-2017-0129

Fu, W., Wang, Q., & Zhao, X. (2018b). Platform-based service innovation and system design: Research opportunities. *Industrial Management & Data Systems*, *118*(5), 975–997. https://doi.org/10.1108/IMDS-03-2017-0130

Fuller, R. B. (1975). *Synergetics: Explorations in the geometry of thinking*. Macmillan.

Furnivall, J. S. (1931). *An Introduction to the Political Economy of Burma*. University of Rangoon.

Furnivall, J. S. (1945a). Some problems of tropical economy. In R. Hinden (Ed.), *Fabian Colonial Essays* (pp. 161–184). George Allen and Unwin.

Furnivall, J. S. (1945b). Some problems of tropical economy. In R. Hinden (Ed.), *Fabian Colonial Essays* (pp. 161–184). George Allen and Unwin.

Furnivall, J. S. (1948a). *Colonial Policy and Practice*. Cambridge University Press.

Furnivall, J. S. (1948b). *Colonial Policy and Practice: A Comparative Study of Burma and Netherlands India*. Cambridge University Press. https://doi.org/10.1017/CBO9781107051140

Gabriel, R. (2007). *The Warrior's Way: A Treatise on Military Ethics*. Canadian Defense Academy Press. http://publications.gc.ca/collections/collection_2012/dn-nd/D2-206-2-2007-eng.pdf

Gaddis, P. (1959). The project manager. Harvard Business Review 37 (3):89-97. *Harvard Business Review*, *37*(3), 89–97.

Gantt, H. L. (1919). *Work, wages, and profits* (2nd ed.). The Engineering Magazine Co. http://archive.org/details/workwagesprofits00gant

Garcia, A. M. M., Verhelle, M., & Vanthienen, J. (2000). *An Overview of decision table literature: 1982-2000*. 132. https://feb.kuleuven.be/Prologa/download/overview82-2000.pdf

Gaudelli, N. M., Komor, A. C., Rees, H. A., Packer, M. S., Badran, A. H., Bryson, D. I., & Liu, D. R. (2017). Programmable base editing of A•T to G•C in genomic DNA without DNA cleavage. *Nature*, *551*(7681), 464–471. https://doi.org/10.1038/nature24644

Genesereth, M., & Fikes, R. (1992). *Knowledge Interchange Format, Version 3.0. Reference Manual*. Interlingua Working Group of the DARPA Knowledge Sharing Effort, with the Computer Science Department of Stanford University. https://www.cs.auckland.ac.nz/courses/compsci367s2c/resources/kif.pdf

Georgescu, I. (2021). 60 years of Landauer's principle. *Nature Reviews Physics*, *3*(12), 12. https://doi.org/10.1038/s42254-021-00400-8

Georgescu-Roegen, N. (1975). Energy and Economic Myths. *Southern Economic Journal*, *41*(3), 347–381.

Geraldi, J. (2012). Gantt charts revisited: A critical analysis of its roots and implications to the management of projects today. *International Journal of Managing Projects in Business*, *5*(4), 578–594. https://doi.org/10.1108/17538371211268889

Gilbert, A., Mesmer, B., & Watson, M. D. (2016). *Uses of Exergy in Systems Engineering*. 12.

Glusker, M., Hogan, D. M., & Vass, P. (2005). The Ternary Calculating Machine of Thomas Fowler. *IEEE Annals of the History of Computing*, *27*(3), 4–22. https://doi.org/10.1109/MAHC.2005.49

Goddard, K., Roudsari, A., & Wyatt, J. C. (2012). Automation bias: A systematic review of frequency, effect mediators, and mitigators. *Journal of the American Medical Informatics Association*, *19*(1), 121–127. https://doi.org/10.1136/amiajnl-2011-000089

Gödel, K. (1931). *On Formally Undecidable Propositions of Principia Mathematica And Related Systems* (B. Meltzer, Trans.). Dover Publications.

Gomes, L. (2018). Quantum computing: Both here and not here. *IEEE Spectrum*, *55*(4), 42–47. https://doi.org/10.1109/MSPEC.2018.8322045

Goñi-Moreno, A., & Amos, M. (2012). A reconfigurable NAND/NOR genetic logic gate. *BMC Systems Biology*, *6*(1), 126. https://doi.org/10.1186/1752-0509-6-126

Goodin, R. E. (2005). Toward an International Rule of Law: Distinguishing International Law-Breakers from Would-Be Law-Makers. *Journal of Ethics*, *9*(1), 225–246.

Goodman, F. (2014). *Algebra: Abstract and Concrete, edition 2.6*. SemiSimple Press (Frederick Goodman).

Government of Canada. (2019, April 23). *Symposium on Algorithmic Government (Agenda)*. https://www.canada.ca/content/dam/ircc/documents/pdf/english/services/agenda-eng.pdf

Gowers, E. (1954). *The Complete Plain Words*. UK Stationary Office.

Grabher, G. (2004). Temporary Architectures of Learning: Knowledge Governance in Project Ecologies. *Organization Studies*, *25*(9), 1491–1514. https://doi.org/10.1177/0170840604047996

Grabher, G., & Ibert, O. (2011). Project Ecologies. In P. Morris, J. Pinto, & J. Söderlund (Eds.), *The Oxford Handbook of Project Management*. http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199563142.001.0001/oxfordhb-9780199563142-e-8

Graeber, C., & Billings, C. (1989). *Human-Centered Automation: Development of a Philosophy*. NASA Ames Research Center. https://ntrs.nasa.gov/api/citations/19910001631/downloads/19910001631.pdf

Grahne, G., & Moallemi, A. (2018). A useful four-valued database logic. *Proceedings of the 22nd*

International Database Engineering & Applications Symposium on - IDEAS 2018, 22–30. https://doi.org/10.1145/3216122.3216157

Greco, G., Liang, F., Palmigiano, A., & Rivieccio, U. (2019). Bilattice logic properly displayed. *Fuzzy Sets and Systems*, *363*, 138–155. https://doi.org/10.1016/j.fss.2018.05.007

Grice, P. (1975). Logic and Conversation. In J. P. Kimball, J. L. Morgan, & P. Cole (Eds.), *Syntax and semantics*. Academic Press, Harcourt Brace Jovanovich.

Grier, D. A. (2000). Agricultural computing and the context for John Atanasoff. *IEEE Annals of the History of Computing*, *22*(1), 48–61. https://doi.org/10.1109/85.815466

Grigg, I. (2004). The Ricardian Contract. *Proceedings of the First IEEE International Workshop on Electronic Contracting*, 25–31. https://doi.org/10.1109/WEC.2004.1319505

Gruber, T. R. (1989). Automated Knowledge Acquisition for Strategic Knowledge. *Machine Learning*, *4*, 293–336.

Gunaratne, R. (1980). The logical form of Catuskoti: A new solution. *Philosophy East and West*, *30*(2), 211–239.

Gunaratne, R. (1986). Understanding Nagarjuna's catuskoti. *Philosophy East and West*, *36*(3), 213–234.

Gupta, R., Asgari, S., Moazamigoodarzi, H., Down, D. G., & Puri, I. K. (2021). Energy, exergy and computing efficiency based data center workload and cooling management. *Applied Energy*, *299*, 117050. https://doi.org/10.1016/j.apenergy.2021.117050

Gurevich, Y. (2014). What is an Algorithm? (Revised). In A. Olszewski (Ed.), *Church's Thesis: Logic, Mind and Nature* (p. 15). Copernicus Center Press.

Gustafson, J. (2000). Reconstruction of the Atanasoff-Berry Computer. In R. Rojas & U. Hashagen (Eds.), *The first computers: History and architectures* (pp. 92–105). MIT Press.

Hage, J. (1999). Moderately Naturalistic Deontic Logic. In P. McNamara & H. Prakken (Eds.), *Norms, Logics and Information Systems: New Studies in Deontic Logic and Computer Science* (pp. 54–72). IOS Press.

Hallé, S. (2022). Test suite generation for boolean conditions with equivalence class partitioning. *Proceedings of the IEEE/ACM 10th International Conference on Formal Methods in Software Engineering*, 23–33. https://doi.org/10.1145/3524482.3527659

Hamano, M. (2012). RNA interference and Register Machines (extended abstract). *Electronic Proceedings in Theoretical Computer Science*, *100*, 107–112. https://doi.org/10.4204/EPTCS.100.8

Hansen, T., Freed, N., & Klensin, J. C. (2005). *RFC 4288. Media Type Specifications and Registration Procedures*. https://tools.ietf.org/html/rfc6838

Harris, R. (2021, March 23). *SvelteKit is in public beta*. https://svelte.dev/blog/sveltekit-beta

Harrison, J. (2014). Soap film solutions to Plateau's problem. *Journal of Geometric Analysis*, *24*(1), 271–297.

Harrison, J., & Pugh, H. (2015). Forthcoming. Plateau's Problem: Existence and Soap Film Regularity of Size-Minimizing Current Solutions. *Journal of Geometric Analysis*. http://math.berkeley.edu/~harrison/Publications_files/HarrisonPughPlateau.pdf

Havelund, K., Ingham, M., & Wagner, D. (2010). *A Case Study in DSL Development*. Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. http://citeseerx.ist.psu.edu/viewdoc/download?

doi=10.1.1.169.637&rep=rep1&type=pdf

Health Canada. (2019). *Post-Notice of Compliance (NOC) Changes: Quality Document*. https://www.canada.ca/content/dam/hc-sc/documents/services/drugs-health-products/drug-products/applications-submissions/guidance-documents/post-notice-compliance-changes/quality-document/quality-document.pdf

Heckmann, I., Comes, T., & Nickel, S. (2015). A critical review on supply chain risk – Definition, measure and modeling. *Omega*, *52*, 119–132. https://doi.org/10.1016/j.omega.2014.10.004

Himanen, P., & Torvalds, L. (2009). *The Hacker Ethic: A Radical Approach to the Philosophy of Business* (Reprint edition). Random House.

Hinum, V., Hanna, A. G., Dario, P., Vadim, T., & Pascal, Z. (2022). *Position Paper: Is the Internet of Rules an appropriate solution to overcome BMW Group's challenges in the area of tariff classification? (Unpublished manuscript)*. University of St.Gallen, School of Management, Economics, Law, Social Sciences, International Affairs and Computer Science.

Hitt, M. A., Beamish, P. W., Jackson, S. E., & Mathieu, J. E. (2007). Building Theoretical and Empirical Bridges Across Levels: Multilevel Research in Management. *Academy of Management Journal*, *50*(6), 1385–1399. https://doi.org/10.5465/amj.2007.28166219

Hjelseth, E., & Nisbet, N. (2010). *Exploring model checking by use of the semantic markup RASE methodology*. 333–340. http://www.baufachinformation.de/kostenlos.jsp?sid=839609EA0B98C5EA233E29F3DFBF4E9C&id=2011071000287&link=http%3A%2F%2Fwww.irbnet.de%2Fdaten%2Ficonda%2FCIB21809.pdf

Hjelseth, E., & Nisbet, N. (2011). *Capturing normative constraints by use of the semantic mark-up RASE methodology*. 11. http://2011-cibw078-w102.cstb.fr/papers/Paper-45.pdf

HL7. (2019a). *Content Logical Definition. Valueset probability distribution type. Fast Healthcase Interoperability Resources (FHIR) v4.0.1*. https://www.hl7.org/fhir/valueset-probability-distribution-type.html

HL7. (2019b). *HL7 Extension: UncertaintyType. Fast Healthcare Interoperability Resources (FHIR) v4.0.1*. https://www.hl7.org/Fhir/extension-iso21090-uncertaintytype-definitions.html#extension.Extension.value[x]

Hoffman, M. D. (2013). Stochastic Variational Inference. *Journal of Machine Learning Research*, *14*, 1303–1347.

Hoffman, W. (2015). *Data-Driven Development*. World Economic Forum. https://www.weforum.org/whitepapers/data-driven-development-pathways-for-progress

Hogg, P. (2007). *Constitutional Law of Canada, 5th Ed.* (1999 student edition). Carswell.

Hohenstein, N., Feisel, E., Hartmann, E., & Giunipero, L. (2015). Research on the phenomenon of supply chain resilience: A systematic review and paths for further investigation. *International Journal of Physical Distribution & Logistics Management*, *45*(1/2), 90–117. https://doi.org/10.1108/IJPDLM-05-2013-0128

Holland, J. (1999). *Emergence: From Chaos to Order*. Da Capo Press.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). *Induction. Processes of inference, learning and discovery*. MIT Press.

Holling, C. S. (1973). Resilience and stability of ecological systems. *Annual Review of Ecology and Systematics*, *4*, 1–23.

Holling, C. S. (1986). The resilience of terrestrial ecosystems: Local surprise and global change. In

W. Clark & Munn (Eds.), *Sustainable Development of the Biosphere* (pp. 293–317). Cambridge University Press, for the International Institute for Applied Systems Analysis (IIASA).

Hong, F., & Šulc, P. (2019). An emergent understanding of strand displacement in RNA biology. *Journal of Structural Biology*, *207*(3), 241–249. https://doi.org/10.1016/j.jsb.2019.06.005

Hopper, G. (1981). Grace Murray Hopper. In L. Gilbert & G. Moore (Eds.), *Particular Passions: Vol. Grace Murray Hopper* (p. 7). Lynn Gilbert Inc. https://www.particularpassions.com/grace-murray-hopper

Hopper, G. (1987). The Education of a Computer. *Annals of the History of Computing. IEEE.*, *9*(3/4), 271–281. https://doi.org/10.1109/MAHC.1987.10032

Horn, A. (1951). On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic*, *16*(1), 14–21. https://doi.org/10.2307/2268661

Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B. N., & Dean, M. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. W3C, for National Research Council of Canada, Network Inference, and Stanford University. https://www.w3.org/Submission/SWRL/

Horváth, I. (2007). Comparison of Three Methodological Approaches of Design Research. *Guidelines for a Decision Support Method Adapted to NPD Processes*. International Conference on Engineering Design, ICED’07 (28 - 31 August 2007), Cité des Sciences et de l’Industrie, Paris, France. http://www.designsociety.org/publication/25512/

Huff, A. S. (2000). Citigroup’s John Reed and Stanford’s James March on management research and practice. *The Academy of Management Executive*, *14*(1), 52–64.

Hume, D. (1738). *A Treatise of Human Nature*. Dover Publications.

Hussein, B., Ngereja, B., Hafseld, K. H. J., & Mikhridinova, N. (2020). Insights on Using Project-Based Learning to Create an Authentic Learning Experience of Digitalization Projects. *2020 IEEE European Technology and Engineering Management Summit (E-TEMS)*, 1–6. https://doi.org/10.1109/E-TEMS46250.2020.9111829

ICCPM. (2012). *Complex Project Manager Competency Standards*. International Centre for Complex Project Management (ICCPM). http://www.defence.gov.au/dmo/proj_man/Complex_PM_v2.0.pdf

IETF. (2010, December 7). *Guidelines to Authors of Internet-Drafts*. IETF. /standards/ids/guidelines/

Ignatova, Z., Martínez-Pérez, I., & Zimmermann, K.-H. (2008). *DNA Computing Models*. Springer US. https://doi.org/10.1007/978-0-387-73637-2

Ihaka, R., & Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, *5*(3), 299–314. https://doi.org/10.1080/10618600.1996.10474713

Indonesia. (2018). *Aircraft Accident Investigation Report. RE: Lion Air flight 610 on October 29, 2018.* Komite Nasional Keselamatan Transportasi. https://reports.aviation-safety.net/2018/20181029-0_B38M_PK-LQP_PRELIMINARY.pdf

Internet Society. (2019). *Global Internet Report 2019 (Consolidation in the Internet Economy)*. Internet Society. https://future.internetsociety.org/2019/

IPMA. (2006). *ICB: IPMA competence baseline ; Version 3.0*. International Project Management Association.

ISNSCE. (2021). *International Conference on DNA Computing and Molecular Programming. International Society of Nanoscale Science, Computing, and Engineering*. http://www.dna-computing.org/

ISO. (1947). *ISO/TC 37—Language and terminology*. ISO. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/committee/04/81/48104.html

ISO. (1986). *Standard Generalized Markup Language (SGML). ISO 8879:1986. Reviewed and confirmed in 2008.* ISO. http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/01/63/16387.html

ISO. (1987). *ISO/TR 9007. Information processing systems — Concepts and terminology for the conceptual schema and the information base*. ISO. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/01/65/16549.html

ISO. (2000). *ISO 1087-1 Terminology work—Vocabulary—Part 1: Theory and application*. ISO. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/02/00/20057.html

ISO. (2001). *ISO/TC 37/SC 4—Language resource management*. ISO. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/committee/29/75/297592.html

ISO. (2009). *ISO 80000-2:2009. Quantities and units—Part 2: Mathematical signs and symbols to be used in the natural sciences and technology*. International Organization for Standardization. http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/18/31887.html

ISO. (2012). *ISO 21500:2012—Guidance on project management*. http://www.iso.org/iso/catalogue_detail.htm?csnumber=50003

ISO. (2014). *ISO 19600:2014*. International Organization for Standardization. http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/23/62342.html

ISO. (2015). *4217—Currency codes*. ISO. https://www.iso.org/iso-4217-currency-codes.html

ISO. (2020). *ISO 3166-2:2020: Codes for the representation of names of countries and their subdivisions—Part 2:Country subdivision code*. International Organization for Standardization. https://www.iso.org/standard/72483.html

ISO/IEC. (2018). *ISO/IEC Directives, Part 2. Principles and Rules for the Structure and Drafting of International Standards, Technical Specifications and Publicly Available Specifications*. International Organization for Standardization and the International Electrotechnical Commission. https://www.iso.org/sites/directives/current/part2/index.xhtml

ISO/IEC. (2019). *ISO 80000-2:2019 Quantities and units—Part 2: Mathematics*. 52.

ITU. (2016). *ITU Radio Regulations—Volume 1*. International Telecommunication Union. https://www.itu.int/pub/R-REG-RR/en

Iyengar, J., & Thomson, M. (2019, July 9). *QUIC: A UDP-Based Multiplexed and Secure Transport. Internet-Draft.* https://tools.ietf.org/html/draft-ietf-quic-transport-22#section-1

James, W. (1922). *Pragmatism, a new name for some old ways of thinking*. Longmans, Green and Co.

Janicki, R. (1995). Towards a formal semantics of Parnas tables. *Proceedings of the 17th International Conference on Software Engineering - ICSE ’95*, 231–240. https://doi.org/10.1145/225014.225036

Janicki, R., Parnas, D. L., & Zucker, J. (1997). Tabular Representations in Relational Documents. In C. Brink, W. Kahl, & G. Schmidt (Eds.), *Relational Methods in Computer Science* (pp. 184–196). Springer Vienna. https://doi.org/10.1007/978-3-7091-6510-2_12

Jayatilleke, K. (1967). The Logic of Four Alternatives. *Philosophy East and West*, *17*(1), 69–83.

Jaynes, E. (1979). Where do we stand on maximum entropy? In R. Levine & M. Tribus (Eds.), *The Maximum Entropy Formalism* (pp. 15–118). MIT Press.

Jeffries, R. (2006). *We Tried Baseball and It Didn't Work*. https://ronjeffries.com/xprog/articles/jatbaseball/

Johnson, S. (1755). *A Dictionary of the English Language* (Facsimile). Times Books. http://archive.org/details/dictionaryofengl01johnuoft

Jonas, W. (2006). Research through design through research: A problem statement and a conceptual sketch. *Design Research Society International Conference, K. Friedman, T. Love, E. Côrte-Real, and C. Rust Eds.* http://www.iade.pt/drs2006/wonderground/proceedings/fullpapers/DRS2006_0230.pdf

Jorgensen, P., & Marselos, N. (1982). *A Modern Appraisal of Decision Tables. Report of the Decision Table Task Group of the Conference on Data Systems Languages (CODASYL)*. Association for Computing Machinery. https://feb.kuleuven.be/Prologa/CODASYL82/CODASYL82.pdf

Kahneman, D. (2011). *Thinking, Fast and Slow*. Doubleday Canada.

Kahneman, D., & Tversky, A. (1982). The simulation heuristic. In *Judgement Under Uncertainty: Heuristics and Biases*. Cambridge University Press.

Kalinowski, J. (1953). Theorie des propositions normatives. *Studia Logica*, *1*, 147–182. https://doi.org/10.1007/BF02272279

Kalyazina, E. (2021). Digital management in project management (Original title: Цифровой менеджмент в управлении проектами) (Deep-L Translator, Trans.). *Jpurnal of Creative Economy*, *15*(12), 4747–4766. https://doi.org/10.18334/ce.15.12.113858

Kapp, E. (1877). *Elements of a Philosophy of Technology: On the Evolutionary History of Culture* (J. West Kirkwood & L. Weatherby, Eds.; L. Wolfe, Trans.; Republished in 2018). U of Minnesota Press.

Kapsali, M. (2011). Systems thinking in innovation project management: A match that works. *International Journal of Project Management*, *29*(4), 396–407. https://doi.org/10.1016/j.ijproman.2011.01.003

Katz, E. (2020). *DNA- and RNA-Based Computing Systems*. John Wiley & Sons.

Kazemzadeh, Y., Milton, S. K., & Johnson, L. W. (2015). Process Chain Network (PCN) and Business Process Modeling Notation (BPMN): A Comparison of Concepts. *Journal of Management and Strategy*, *6*(1), 12.

Keen, P. G. W., & Scott Morton, M. S. (1978). *Decision support systems: An organizational perspective*. Reading, Mass. : Addison-Wesley Pub. Co. http://archive.org/details/decisionsupports00keen_0

Kelly, D. (2020, May 18). *Direct email correspondence about the expression of a sample rule in PASCAL* [Personal communication].

Kelly, D. (2022, March 3). *Work-in-Progress Demo of a RuleTaker and RuleReserve Reference Implementation*.

Kelter, J., Conboy, W., Potvin, J., & Wilensky, U. (2022). *A Macroeconomic Agent-based Modeling Framework with Arbitrary Supply Network Complexity (Poster Submission)*. Computational Social Science Society of the Americas (CSS2021) Annual Conference, Online. http://ccl.northwestern.edu/papers.shtml

Kelter, J., Wilensky, U., & Potvin, J. (2022). Introducing Land Constraints to Macroeconomic Agent-based Models. *Proceedings of the 2022 Conference of The Computational Social Science Society of the Americas, Forthcoming*. http://ccl.northwestern.edu/papers.shtml

Keynes, J. (1921). *A Treatise On Probability* (Unabridged edition). Cambridge University Press.

Khalid, M., & Singh, J. (2016). Memristor based unbalanced ternary logic gates. *Analog Integrated Circuits and Signal Processing*, *87*(3), 399–406. https://doi.org/10.1007/s10470-016-0733-1

Khan, M. H. A. (2008). Reversible Realization of Quaternary Decoder, Multiplexer, and Demultiplexer Circuits. *38th International Symposium on Multiple Valued Logic (Ismvl 2008)*, 208–213. https://doi.org/10.1109/ISMVL.2008.33

Kifer, M., & Boley, H. (2013, February 5). *Rule Interchange Format (RIF) Overview (Second Edition)*. https://www.w3.org/TR/rif-overview/

Killen, C., Jugdev, K., Drouin, N., & Petit, Y. (2012). Advancing project and portfolio management research: Applying strategic management theories. *International Journal of Project Management*, *30*(5), 525–538. https://doi.org/10.1016/j.ijproman.2011.12.004

Kim, H., Bojar, D., & Fussenegger, M. (2019). A CRISPR/Cas9-based central processing unit to program complex logic computation in human cells. *Proceedings of the National Academy of Sciences*, *116*(15), 7214–7219. https://doi.org/10.1073/pnas.1821740116

Kingston, G., Henderson, D., & Vernik, R. (1999). *An Approach for Identifying and Characterising Problems in the IterativeDevelopment of C3I Capability*. Defense Science and Technology Organisation, Australia. http://www.dtic.mil/dtic/tr/fulltext/u2/a375433.pdf

Kloppenberg, J. T. (1996). Pragmatism: An Old Name for Some New Ways of Thinking? *The Journal of American History*, *83*(1), 100. https://doi.org/10.2307/2945476

Knight, W. (2019). *How malevolent machine learning could derail AI*. MIT Technology Review. https://www.technologyreview.com/s/613170/emtech-digital-dawn-song-adversarial-machine-learning/

Kogut, B., & Kulatilaka, N. (1994). Options Thinking and Platform Investment: Investing in Opportunity. *California Management Review*, *36*(2), 52–71.

Korzun, D., & Gurtov, A. (2013). *Structured Peer-to-Peer Systems*. Springer New York. https://doi.org/10.1007/978-1-4614-5483-0

Kowalski, R. (1974). Predicate Logic as Programming Language. *Proceedings of the IFIP Congress*, 569–574.

Kowalski, R. (1979a). Algorithm = Logic + Control. *Communications of the ACM*, *22*(7), 424–436.

Kowalski, R. (1979b). *Logic for problem solving*. Elsevier North Holland.

Kozarkiewicz, A. (2020). General and Specific: The Impact of Digital Transformation on Project Processes and Management Methods. *Foundations of Management*, *12*(1), 237–248. https://doi.org/10.2478/fman-2020-0018

Kripke, S. A. (1982). *Wittgenstein on Rules and Private Language: An Elementary Exposition*. Harvard University Press.

Kuechler, W., & Vaishnavi, V. (2012). A Framework for Theory Development in Design Science

Research: Multiple Perspectives. *Journal of the Association for Information Systems*, *13*(6), 395–423.

Kuhn, T. (2014). A Survey and Classification of Controlled Natural Languages. *Computational Linguistics*, *40*(1), 121–170. https://doi.org/10.1162/COLI_a_00168

Kuhn, T. S. (1962). *The structure of scientific revolutions*. The University of Chicago Press.

Kuhn, T. S. (1970). Reflections on My Critics. In A. Musgrave & I. Lakatos (Eds.), *Criticism and the Growth of Knowledge. Proceedings of the International Colloquium in the Philosophy of Science* (Vol. 4, pp. 231–277). Cambridge University Press.

Kupperman, J. J. (2005). A New Look at the Logic of the Is-Ought Relation. *Philosophy*, *80*(03), 343. https://doi.org/10.1017/S0031819105000331

Kurthy, M., Lawford-Smith, H., & Sousa, P. (2017). Does ought imply can? *PLoS ONE*, *12*(4), 24. https://doi.org/10.1371/journal. pone.0175206

Lalonde, P., Bourgault, M., & Findeli, A. (2010). Building pragmatist theories of PM practice: Theorizing the act of project management. *Project Management Journal*, *41*(5), 21–36. https://doi.org/10.1002/pmj.20163

Lalonde, P.-L., Bourgault, M., & Findeli, A. (2010). Building pragmatist theories of PM practice: Theorizing the act of project management. *Project Management Journal*, *41*(5), 21–36. https://doi.org/10.1002/pmj.20163

Landauer, R. (1967). Wanted: A Physically Possible Theory of Physics. *IEEE Spectrum*, *4*(9), 105–109.

Latham, R. (2000). Social Sovereignty. *Theory, Culture & Society*, *17*(4), 1–18.

Lawvere, W. (1969). Diagonal arguments and cartesian closed categories. *Category Theory, Homology Theory and Their Applications*, *2*, 134–145.

Leach, P. J., Salz, R., & Mealling, M. H. (2005). *A Universally Unique IDentifier (UUID) URN Namespace* (Request for Comments RFC 4122). Internet Engineering Task Force. https://doi.org/10.17487/RFC4122

Lee, C. (2018, June 28). *There are more ways to arrange a deck of cards than there are atoms on Earth*. McGill University, Office for Science and Society. https://www.mcgill.ca/oss/article/did-you-know-infographics/there-are-more-ways-arrange-deck-cards-there-are-atoms-earth

Lehr, W., Clark, D., Bauer, S., Berger, A., & Richter, P. (2019). Whither the Public Internet? *Journal of Information Policy*, *9*, 1–42. https://doi.org/10.5325/jinfopoli.9.2019.0001

Lei, Z., & Coulton, P. (2011). Using Deliberate Ambiguity of the Information Economy in the Design of a Mobile Location Based Games. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, 33–36. https://doi.org/10.1145/2181037.2181044

Leonard, A. (2015). Stafford Beer and the legacy of Cybersyn: Seeing around corners. *Kybernetes*, *44*(6/7), 926–934. https://doi.org/10.1108/K-02-2015-0045

Levitt, T. (1991). *Thinking About Management (First Edition)*. Free Press.

Lew, A. (1982). On the emulation of flowcharts by decision tables. *Communications of the ACM*, *25*(12), 895–905. https://doi.org/10.1145/358728.358739

Lew, A. (1983). Decision tables for general-purpose scientific programming. *Software: Practice and Experience*, *13*(2), 181–188. https://doi.org/10.1002/spe.4380130207

Lew, A., & Tamanaha, D. (1976). Decision Table Programming and Reliability. *ICSE '76: Proceedings of the 2nd International Conference on Software Engineering*, 345–349.

Li, D. X. (2000). On Default Correlation: A Copula Function Approach. *Journal of Fixed Income*, *9*, 43–54.

Li, Y., Liu, W., Cao, B., Yin, J., & Yao, M. (2016). An efficient MapReduce-based rule matching method for production system. *Future Generation Computer Systems*, *54*, 478–489. https://doi.org/10.1016/j.future.2015.03.010

Liew, A. (2013). DIKIW: Data, Information, Knowledge, Intelligence, Wisdom and their Interrelationships. *Business Management Dynamics*, *2*(10), 49–62.

Lima, L. G., Soares-Neto, F., Lieuthier, P., Castor, F., Melfe, G., & Fernandes, J. P. (2016). Haskell in Green Land: Analyzing the Energy Behavior of a Purely Functional Language. *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 517–528. https://doi.org/10.1109/SANER.2016.85

Limnios, E., Mazzarol, T., Ghadouani, A., & Schilizzi, S. (2014). The Resilience Architecture Framework: Four organizational archetypes. *European Management Journal*, *32*(1), 104–116. https://doi.org/10.1016/j.emj.2012.11.007

Lin, S., & Huang, D. (2020). *Project Management Under Internet Era*. Springer Singapore. https://doi.org/10.1007/978-981-15-2799-9

Lindberg, V. (2018, October 18). *MongoDB's Server Side Public License is Likely Unenforceable*. Process Mechanics. https://www.processmechanics.com/2018/10/18/the-server-side-public-license-is-flawed/

Lipton, R., & Baum, E. (Eds.). (1996). *DNA Based Computers* (Vol. 27). American Mathematical Society. https://doi.org/10.1090/dimacs/027

Litman, J. (2003). Ethical disobedience. *Ethics and Information Technology*, *5*(4), 217–223. https://doi.org/10.1023/B:ETIN.0000017736.38811.22

Liu, Q., Yang, K., Xie, J., & Sun, Y. (2022). DNA-Based Molecular Computing, Storage, and Communications. *IEEE Internet of Things Journal*, *9*(2), 897–915. https://doi.org/10.1109/JIOT.2021.3083663

Lloyd, E. (1997). Feyerabend, Mill, and Pluralism. *Philosophy of Science*, *64*(1), 396–407. https://doi.org/10.1086/392617

Lloyd, J. W. (1987). *Foundations of Logic Programming*. Springer Science & Business Media.

Lobaugh, N. J., Cole, S., & Rovet, J. F. (1998). Visual search for features and conjunctions in development. *Canadian Journal of Experimental Psychology/Revue Canadienne de Psychologie Expérimentale*, *52*(4), 201–212. https://doi.org/10.1037/h0087293

Lotka, A. J. (1925). *Elements of Physical Biology—Scholar's Choice Edition* (Republished in 2015). Creative Media Partners, LLC.

Lovelock, J. E. (1993). The soil as a model for the Earth. *Geoderma*, *57*(3), 213–215. https://doi.org/10.1016/0016-7061(93)90003-4

Luehrman, T. (1998). Strategy as a Portfolio of Real Options. *Harvard Business Review*, *76*(5), 89–99.

Łukasiewicz, J. (1920). On three-valued logic. (Original in Polish: O logice trójwartościowej. Ruch filozoficzny 5. Pp. 170–171). In L. Borkowski (Ed.), *Selected works by Jan Łukasiewicz* (pp. 87–88). North–Holland.

Lukka, K., & Suomala, P. (2014). Relevant interventionist research: Balancing three intellectual virtues. *Accounting and Business Research*, *44*(2), 204–220. https://doi.org/10.1080/00014788.2013.872554

Ma, D., Li, Y., Wu, K., Yan, Z., Tang, A. A., Chaudhary, S., Ticktin, Z. M., Alcantar-Fernandez, J., Moreno-Camacho, J. L., Campos-Romero, A., & Green, A. A. (2022). Multi-arm RNA junctions encoding molecular logic unconstrained by input sequence for versatile cell-free diagnostics. *Nature Biomedical Engineering*, *6*(3), 298–309. https://doi.org/10.1038/s41551-022-00857-7

Madanayake, R., Dias, G., & Kodikara, N. (2015). A Discovery of the Relevance of Eastern Four-valued (Catuskoti) Logic to Define Modular Transformations When There are Multiple Ways of Representing the Same Modular Transformation. *Computer Science and Information Technology*, *3*(6), 247–255. https://doi.org/10.13189/csit.2015.030603

Magnusson, J., Elliot, V., & Hagberg, J. (2022). Digital transformation: Why companies resist what they need for sustained performance. *Journal of Business Strategy*, *43*(5), 316–322. https://doi.org/10.1108/JBS-02-2021-0018

Maher, D. W., & Makowski, J. F. (2001). Literary Evidence for Roman Arithmetic with Fractions. *Classical Philology*, *96*(4), 376–399. https://doi.org/10.1086/449557

Mahgoub, A., Ganesh, S., Meyer, F., Grama, A., & Chaterji, S. (2017). Suitability of NoSQL systems—Cassandra and ScyllaDB — For IoT workloads. *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, 476–479. https://doi.org/10.1109/COMSNETS.2017.7945437

Maier, D., Tekele, K. T., Kifer, M., & Warren, D. (2018). Datalog: Concepts, History and Outlook. In M. Kifer & Y. Liu (Eds.), *Declarative Logic Programming: Theory, Systems, and Applications* (pp. 3–100). Morgan & Claypool.

Maitrey, S., & Jha, C. K. (2015). MapReduce: Simplified Data Analysis of Big Data. *Procedia Computer Science*, *57*, 563–571. https://doi.org/10.1016/j.procs.2015.07.392

Malcolm, N. (1989). Wittgenstein on Language and Rules. *Philosophy*, *64*(247), 5–28. JSTOR.

Maranzana, N., Gartiser, N., & Caillaud, E. (2008). *From concurrent engineering to collaborative learning of design*. *4*, 14.

Markovits, D. (2016). Civility, Rule-Following and the Authority of Law. *Columbia Law Review*, *116*. https://columbialawreview.org/content/civility-rule-following-and-the-authority-of-law/

Marston, T. (2001, December 5). *Building a Rule Based Calculation Engine with Customisable Formulae*. https://www.tonymarston.net/uniface/calculationengine.html

Masinter, L. M., Zigmond, D., Alvestrand, H. T., & Petke, R. A. (1999). *Guidelines for new URL Schemes* (Request for Comments RFC 2718). Internet Engineering Task Force. https://doi.org/10.17487/RFC2718

Mathy, F., & Feldman, J. (2012). What's magic about magic numbers? Chunking and data compression in short-term memory. *Cognition*, *122*(3), 346–362. https://doi.org/10.1016/j.cognition.2011.11.003

Matsuura, S., Ono, H., Kawasaki, S., Kuang, Y., Fujita, Y., & Saito, H. (2018). Synthetic RNA-based logic computation in mammalian cells. *Nature Communications*, *9*(1), 4847. https://doi.org/10.1038/s41467-018-07181-2

Maturana, H. R., & Varela, F. J. (1980). *(Reprinted in 2012) Autopoiesis and Cognition: The Realization of the Living*. Springer Science & Business Media.

Maxwell, N. (1972). A Critique of Popper’s Views on Scientific Method. *Philosophy of Science*, *39*(2), 131–152.

McDonough, A. M. (1963). *Information Economics and Management Systems*. McGrawHill.

McDonough, A. M. (Ed.). (1987). *Critical Issues in NASA Information Systems: Final Report to the National Aeronautics and Space Administration*. National Academy Press. https://doi.org/10.17226/10447

McGrath, J., & Kostalova, J. (2020). *Project Management Trends and New Challenges 2020+* (P. Maresova, P. Jedlicka, K. Firlej, & I. Soukal, Eds.; pp. 534–542). https://doi.org/10.36689/uhk/hed/2020-01-061

McIlroy, D. (2009, March 12). *Presentation to the Dartmouth-Lake Sunapee Linux User Group (DLSLUG)*.

McIlroy, D., Pinson, E., & Tague, B. (1978). UNIX Time-Sharing System. *Bell System Technical Journal*, *57*(6), 1899–1904.

Medina, E. (2011). *Cybernetic revolutionaries: Technology and politics in Allende’s Chile*. MIT Press.

Medina, E. (2015). Rethinking algorithmic regulation. *Kybernetes*, *44*(6/7), 1005–1019. https://doi.org/10.1108/K-02-2015-0052

Melnikov, A., & Hansen, T. (2013). *RFC 6839. Additional Media Type Structured Syntax Suffixes*. https://tools.ietf.org/html/rfc6839

Mernik, M., Heering, J., & Sloane, A. M. (2005). When and how to develop domain-specific languages. *ACM Computing Surveys*, *37*(4), 316–344. https://doi.org/10.1145/1118890.1118892

Meyer, J. W., & Rowan, B. (1977). Institutionalized Organizations: Formal Structure as Myth and Ceremony. *American Journal of Sociology*, *83*(2), 340–363.

Mill, J. S., & Mill, H. T. (1863). *On Liberty*. Ticknor and Fields.

Miller, A., & Wright, C. (Eds.). (2002). *Rule-Following and Meaning*. Carleton University Press.

Miller, G. A. (1994). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, *101*(2), 345–352.

Minsky, M. (1967). *Prentice-Hall Series in Automatic Computation*. Prentice Hall.

Misra, V. (2015). Routing money, not packets: Revisiting Network Neutrality. *Communications of the ACM*, *58*(6), 24–27. https://doi.org/10.1145/2753120

Mitchell, C. M. (1996). Human-Centered Automation: A Philosophy, Some Design Tenets, and Related Research. In C. A. Ntuen & E. H. Park (Eds.), *Human Interaction with Complex Systems: Conceptual Principles and Design Practice* (pp. 377–381). Springer US. https://doi.org/10.1007/978-1-4613-1447-9_31

Mitsikas, T. (2022, April 12). *Direct email correspondence about the expression of a sample rule in RuleML* [Personal communication].

Mizas, Ch., Sirakoulis, G. Ch., Mardiris, V., Karafyllidis, I., Glykos, N., & Sandaltzopoulos, R. (2008). Reconstruction of DNA sequences using genetic algorithms and cellular automata: Towards mutation prediction? *Biosystems*, *92*(1), 61–68. https://doi.org/10.1016/j.biosystems.2007.12.002

Mladenic, D., Lavrač, N., Bohanec, M., & Moyle, S. (2003). *Data Mining and Decision Support: Integration and Collaboration*. Springer Science & Business Media.

Moaiyeri, M. H., Navi, K., & Hashemipour, O. (2012). Design and Evaluation of CNFET-Based Quaternary Circuits. *Circuits, Systems, and Signal Processing*, *31*(5), 1631–1652. https://doi.org/10.1007/s00034-012-9413-2

Mohun, J., & Roberts, A. (2020). *Cracking the Code: Rulemaking for humans and machines* (OECD Working Papers on Public Governance No. 42; OECD Working Papers on Public Governance, Vol. 42). https://doi.org/10.1787/3afe6ba5-en

Moktefi, A., & Shin, S.-J. (2012). A History of Logic Diagrams. In *Handbook of the History of Logic* (Vol. 11, pp. 611–682). Elsevier. https://doi.org/10.1016/B978-0-444-52937-4.50011-3

Moon, T. S., Lou, C., Tamsir, A., Stanton, B. C., & Voigt, C. A. (2012). Genetic programs constructed from layered logic gates in single cells. *Nature*, *491*(7423), 249–253. https://doi.org/10.1038/nature11516

Moore, G., & Courtland, R. (2015, March 30). Gordon Moore: The Man Whose Name Means Progress. *IEEE Spectrum*. https://spectrum.ieee.org/gordon-moore-the-man-whose-name-means-progress

Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics*, *38*(8), 4.

Mori, K. (1984). Proposition of autonomous decentralized concept. *Transactions of the IEEE, 104C, No. 12*, 303–340.

Mori, K. (1993). Autonomous Decentralized Systems: Concepts, data field architecture and future trends. *Proceedings of the IEEE Conf. on ISADS*, 28–34.

Mori, K. (2007). Autonomous Decentralized System for Service Assurance and Its Application. In M. Malek, M. Reitenspieß, & A. van Moorsel (Eds.), *Service Availability* (pp. 1–8). Springer. https://doi.org/10.1007/978-3-540-72736-1_1

Morris, P., & Geraldi, J. (2011). Managing the institutional context for projects. *Project Management Journal*, *42*(6), 20–32. https://doi.org/10.1002/pmj.20271

Moschovakis, Y. N. (2001). What is an Algorithm? In B. Engquist & W. Schmid (Eds.), *Mathematics Unlimited: 2001 and Beyond* (pp. 919–936). Springer.

Mosier, K. L., Skitka, L. J., Burdick, M. D., & Heers, S. T. (1996). Automation Bias, Accountability, and Verification Behaviors. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *40*(4), 204–208. https://doi.org/10.1177/154193129604000413

Müller, R., Martinsuo, M., & Blomquist, T. (2008). Project portfolio control and portfolio management performance in different contexts. *Project Management Journal*, *39*(3), 28–42. https://doi.org/10.1002/pmj.20053

Munasinghe, M., & McNeely, J. (1995). Key Concepts and Terminology of Sustainable Development Mohan Munasinghe and Jeffrey McNeely. In M. Munasinghe & W. Schearer (Eds.), *Defining and measuring sustainability: The biogeophysical foundations* (pp. 19–56). The World Bank, for the United Nations University. https://www.researchgate.net/publication/246603850_Key_concepts_and_terminology_of_sustainable_development

Murray, C. (2015). Fifty Shades Of Red: A Modest Proposal For Rejecting Rules. *Dow Jones Institutional New (Reprinted from the Wall Stree Journal, 8 May, 2015*, 4.

Nadeem, S. (2019, March 29). The Deadly Price of the Automation Paradox. *The Walrus*. https://thewalrus.ca/the-deadly-price-of-the-automation-paradox/

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. 9.

Nature. (2014). Software with impact. *Nature Methods*, *11*, 211.

Naur, P. (1977). The Science of Informatics and Its Applications. In R. Nossum (Ed.), *Informatics* (pp. 3–17). North-Holland.

Navidi, A., Sabbaghi-Nadooshan, R., & Dousti, M. (2021). A creative concept for designing and simulating quaternary logic gates in quantum-dot cellular automata. *Frontiers of Information Technology & Electronic Engineering*, *22*(11), 1541–1550. https://doi.org/10.1631/FITEE.2000590

Nedelkoska, L., & Quintini, G. (2018). *Automation, skills use and training* (OECD Social, Employment and Migration Working Papers No. 202; OECD Social, Employment and Migration Working Papers, Vol. 202). https://doi.org/10.1787/2e2f4eea-en

Nemirovski, D. (2021). *Utilization of Elements of Digital Transformation in Project Management: A comparative study*. 18–30. http://www.balticpmconference.eu/sites/default/files/image-uploads/Nemirovski%20Dimitri%20%28France%29.%20Utilization%20of%20Elements%20of%20Digital%20Transformation%20in%20Project%20Management.pdf

Newby, G. B. (2001). Cognitive space and information space. *Journal of the American Society for Information Science and Technology*, *52*(12), 1026–1048. https://doi.org/10.1002/asi.1172

Neyland, D., & Möllers, N. (2017). Algorithmic IF … THEN rules and the conditions and consequences of power. *Information, Communication & Society*, *20*(1), 45–62. https://doi.org/10.1080/1369118X.2016.1156141

Ngereja, B. J., & Hussein, B. (2021). An examination of the preconditions of learning to facilitate innovation in digitalization projects: A project team members' perspective. *International Journal of Information Systems and Project Management*, *9*(2), 23–41. https://doi.org/10.12821/ijispm090202

Nicolini, D. (2012). *Practice Theory, Work, and Organization: An Introduction*. OUP Oxford.

Nicolis, G., & Prigogine, I. (1989). *Exploring complexity: An introduction*. W.H. Freeman.

Nielsen, M., & Chuang, I. (2010). *Quantum Computation and Quantum Information, 10th Anniversary Edition*. Cambridge University Press.

Norman, D. (2013). Basic Books.

North, D. (2006, September 20). Introducing BDD (Behaviour-Driven Development). Updated from original version in: Better Software Magazine. *Dan North & Associates*. https://dannorth.net/introducing-bdd/

Nyholm, S. (2018). Attributing Agency to Automated Systems: Reflections on Human–Robot Collaborations and Responsibility-Loci. *Science and Engineering Ethics*, *24*(4), 1201–1219. https://doi.org/10.1007/s11948-017-9943-x

OAG Canada. (2018, May 29). *Report 1—Building and Implementing the Phoenix Pay System*. Office of the Auditor General of Canada. https://www.oag-bvg.gc.ca/internet/English/att__e_43045.html

OECD. (2016). *Big Data: Bringing Competition Policy to the Digital Era. Background note by the Secretariat, DAF/COMP(2016)14, Version dated 2016-10-27.* Organisation for Economic Co-operation and Development, Directorate for Financial and Enterprise Affairs, Competition Committee. http://www.oecd.org/daf/competition/big-data-bringing-competition-policy-to-the-digital-era.htm

OGC. (2019). *OGC Testbed-14: Compliance Engineering Report*. http://docs.opengeospatial.org/per/18-034r3.html#WFSTest

OMG. (2005). *About the Semantics Of Business Vocabulary And Rules Specification*. https://www.omg.org/spec/SBVR/About-SBVR/

OMG. (2013, July). *ISO/IEC 19510:2013. Information technology—Object Management Group Business Process Model and Notation (BPMN)*. ISO. http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/26/62652.html

OMG. (2016a). *About the Requirements Interchange Format Specification Version 1.2*. https://www.omg.org/spec/ReqIF/About-ReqIF/

OMG. (2016b). *The RuleSpeak Business Rule Notation. Semantics of Business Vocabulary and Business Rules (SBVR), v1.4*. Object Management Group. https://www.omg.org/cgi-bin/doc?formal/19-10-07.pdf

OMG. (2016c). *The RuleSpeak Business Rule Notation. Semantics of Business Vocabulary and Business Rules (SBVR), v1.4*. Object Management Group. https://www.omg.org/cgi-bin/doc?formal/19-10-07.pdf

OMG. (2019a). *Decision Model and Notation Version 1.3*. Object Management Group. https://www.omg.org/spec/DMN/

OMG. (2019b). *Semantics of Business Vocabulary and Business Rules, Version 1.5 (First published in 2008)*. Object Management Group. https://www.omg.org/spec/SBVR/1.5/PDF

Onishi, T. (2015). The Catuskoti in a Bilattice: Prospectus. *Journal of the Laboratory of Philosophy (자는이정표정준형입니다) , Kyoto University*, *18*, 19–30.

Ontario. (2021, November 27). *Licensing. Ontario Regulation 746/21 made under the Liquor Licence and Control Act, 2019. Printed in The Ontario Gazette.* https://www.ontario.ca/laws/view

Open Source Initiative. (1998). *The Open Source Definition*. http://opensource.org/osd

OpenAI. (2022, November 14). *OpenAI Sharing & Publication Policy. OpenAI API Terms and Policies.* OpenAI. https://openai.com/api/policies/sharing-publication/

OpenAI Inc. (2021, March 25). *GPT-3 Powers the Next Generation of Apps*. OpenAI. https://openai.com/blog/gpt-3-apps/

Orlowski, J. (2020). *The Social Dilemma* [Documentary]. Netflix. https://www.thesocialdilemma.com/

Ormerod, R. (2009). Rational inference: Deductive, inductive and probabilistic thinking. *Journal of the Operational Research Society*. https://doi.org/10.1057/jors.2009.96

Ostrom, E. (1990). *Governing the Commons*. Cambridge University Press.

Ostrom, E., & Crawford, S. (2005). Chapter 5: A Grammar of Institutions. In *Understanding institutional diversity*. Princeton University Press.

Overell, P., & Crocker, D. (2008). *RFC5234. Augmented BNF for Syntax Specifications: ABNF (Replaces RFC2234 originally published in 1997.)*. https://tools.ietf.org/html/rfc5234

Padmanabhan, M., Martin, K., & Péceli, G. (1996). *Feedback-Based Orthogonal Digital Filters* (Vol. 343). Springer US. https://doi.org/10.1007/978-1-4613-1305-2

Panti, G. (1998). Multi-Valued Logics. In P. Smets (Ed.), *Quantified Representation of Uncertainty and Imprecision* (pp. 25–74). Springer Netherlands. https://doi.org/10.1007/978-94-017-1735-9_2

Parnas, D., Madey, J., & Iglewski, M. (1994). Precise documentation of well-structured programs. *IEEE Transactions on Software Engineering*, *20*(12), 948–976. https://doi.org/10.1109/32.368133

Parry, R., Poole, N., & Pratty, J. (2010). Semantic Dissonance: Do we need (and do we understand) the semantic Web? In R. Parry (Ed.), *Museums in a Digital Age* (pp. 96–106). Routledge.

Parviainen, P., Tihinen, M., Kääriäinen, J., & Teppola, S. (2017). Tackling the digitalization challenge: How to benefit from digitalization in practice. *International Journal of Information Systems and Project Management*, *5*(1), 63–77. https://doi.org/10.12821/ijispm050104

Pascalau, E., & Giurca, A. (2009). JSON Rules—The JavaScript Rule Engine. *Technical Report No. 461, Wurzburg University*, 63–65. http://ceur-ws.org/Vol-486/kese2009-proceedings.pdf

Patterson, D. A., & Ditzel, D. R. (2000). The Case for the Reduced Instruction Set Computer. *Readings in Computer Architecture*, *January*, 135–143.

Peirce, C. S. (1885). On the algebra of logic: A contribution to the philosophy of notation. *American Journal of Mathematics*, *7*(2), 180–202.

Peirce, C. S., & Marquand, A. (1883). *Studies in Logic*. Little, Brown, and Company.

Peñalver, E. M., & Katyal, S. (2010). *Property Outlaws: How Squatters, Pirates, and Protesters Improve the Law of Ownership*. Yale University Press.

Petit, Y., & Hobbs, B. (2010). Project portfolios in dynamic environments: Sources of uncertainty and sensing mechanisms. *Project Management Journal*, *41*(4), 46–58. https://doi.org/10.1002/pmj.20201

Phillips, A., & Cardelli, L. (2009). A programming language for composable DNA circuits. *Journal of The Royal Society Interface*, *6*(suppl_4). https://doi.org/10.1098/rsif.2009.0072.focus

Phillips, A., & Davis, M. (2009). *Tags for Identifying Languages* (Request for Comments RFC 5646). Internet Engineering Task Force. https://doi.org/10.17487/RFC5646

Plank, B. (2016). *What to do about non-standard (or non-canonical) language in NLP*. Unpublished research discussion paper available via Cornell University Library: arXiv digital archive. http://arxiv.org/abs/1608.07836

Plateau, J. (1873). *Experimental and theoretical statics of liquids subject to molecular forces only*. Gauthier- Villars.

Plikynas, D., & Raudys, S. (2015). Towards Nonlocal Field-Like Social Interactions: Oscillating Agent Based Conceptual and Simulation Framework. In D. Secchi & M. Neumann (Eds.), *Agent-Based Simulation of Organizational Behavior: New Frontiers of Social Science Research* (pp. 237–263). Springer.

Plötz, T., Wannenwetsch, A. S., & Roth, S. (2018). Stochastic Variational Inference with Gradient Linearization. *ArXiv:1803.10586 [Cs, Stat]*. http://arxiv.org/abs/1803.10586

Plutarch. (100 C.E.). *Parallel Lives of Greeks and Romans. 1917 Loeb Classical Library Edition, transl. From the original.* https://archive.org/details/parallel_lives01_0810_librivox1

PMI. (2021). *Pulse of the Profession 2021*. Project Management Institute. https://www.pmi.org/learning/thought-leadership/pulse/pulse-of-the-profession-2021

Popper, K. R. (Karl R. (1979). *Objective knowledge: An evolutionary approach*. Oxford [Eng.] : Clarendon Press ; New York : Oxford University Press. http://archive.org/details/objectiveknowled00popp

Post, E. L. (1921). Introduction to a General Theory of Elementary Propositions. *American Journal of Mathematics*, *43*(3), 163. https://doi.org/10.2307/2370324

Potvin, J. (1986). *Economic theory and development planning in the Arctic regions: A study in plural economy* [Thesis, University of Cambridge]. https://doi.org/10.17863/CAM.80819

Potvin, J. (1992). *Classification and Appraisal Criteria for Conservation Investments: A proposed general framework.* Unpublished working paper prepared under contract with The World Bank, on behalf of the Global Environment Facility.

Potvin, J. (Ed.). (2014a). *Free/Libre/Open Works (FLOW): A Syllabus for Professionals, assembled by the Management Education Working Group*. Open Source Initiative, with sponsorship from Ericsson Inc. http://osi.xwiki.com/bin/Projects/FreeLibreOpenWorksManagementEducation

Potvin, J. (2014b). *Restricted vs Free Market Pricing in Web Payments: Considerations in accommodating transparency and freedom of choice in "unit-of- account", in "medium-of-exchange" and in "value-in-exchange benchmarking" at the inception of a possible W3C web payments specification*. W3C Workshop on Web Payments: How do you want to pay?, Paris. http://www.w3.org/2013/10/payments/

Potvin, J. (2014c). *The Methodological Significance of Uncertainty*. Open Source Initiative.

Potvin, J. (2015, May 26). *A Quick Introduction to UBL Oriented to Payment Solutions Designers*. W3C, Web Payments Community Group. https://www.w3.org/community/webpayments/wiki/A_Quick_Introduction_to_UBL_Oriented_to_Payment_Solutions_Designers

Potvin, J. (2018). *Silos ⟶ Nodes: How can silos that prevent interoperability be transformed into nodes of a network? Facilitator's Report of a Round Table Session.* World Trade Symposium 2018 (Open Markets, Open Finance, Open Architecture), London.

Potvin, J. (2019a). *Earth Reserve Assurance: A Sound Money Framework. Pre-submission request for comment, Version 0.6.1 Article in partial fulfilment of a Doctorate in Administration (Project Management). Université du Québec. Canada.* https://drive.google.com/file/d/1F5xtzrjX61VmLUOv3TziPgyR5GKZdA87/view?usp=sharing

Potvin, J. (2019b). *An Experiment in Automated Discovery of "In Effect" and "Applicable" Technical Standards Requirements During Unit Testing. Draft Proposal to the IETF (General Area; RFC-Editor)*. Unpublished proposal from Xalgorithms Foundation.

Potvin, J. (2021, January 29). *How to Scale DWDS Deployment. Reply to a question from the Deputy Director of Digital Governance in a national government.* [Personal communication].

Potvin, J. (2023). *Data with Direction: Design Research Leading to a System Specification for 'an Internet of Rules'. Submitted to the Board of Examiners in partial fulfillment of a Doctorate in Business Administration—DBA (Project Management). License CC-by 4.0* [Université du Québec—Outaouais Campus (UQO)]. https://gitlab.com/xalgorithms-alliance/data-with-direction-specification/dwds-documents/-/tree/master/current

Potvin, J., & OpenAI/GPT-3. (2022, December). *A text-based discussion between a human and a machine on elements of applied and philosophical logic.* [Interview]. https://beta.openai.com/playground

Pradilla Rueda, M. (2008). *Vers une épistémologie de la théorie informatique* [Thèse de doctorat en Philosophie, l'université Paris 1 Panthéon-Sorbonne]. https://www.theses.fr/2008PA010579

Pradilla Rueda, M. (2017). *Lógica Basica: Reflexiónes epistemológicas, históricas y filosóficas, Vol. 2, Pt 1*. Corporación Universitaria Republicana.

Preston-Werner, T. (2013). *Semantic Versioning (SemVer) 2.0.0*. Semantic Versioning. https://semver.org/spec/v2.0.0.html

Priest, G. (2006a). *In Contradiction: A Study of the Transconsistent*. Oxford University Press.

Priest, G. (2006b). *In Contradiction*. Clarendon Press.

Priest, G. (2010). The Logic of Catuskoti. *Comparative Philosophy*, *1*(2). https://doi.org/10.31979/2151-6014(2010).010206

Priest, G. (2014). Beyond True and False. *Aeon*. https://aeon.co/essays/the-logic-of-buddhist-philosophy-goes-beyond-simple-truth

Priest, G. (2018). *The fifth corner of four: An essay on buddhist metaphysics and the catuṣkoṭi* (First edition). Oxford University Press.

Priest, G. (2022). *Brief question and answer exchange regarding various logic frameworks, initiated by Joseph Potvin beween 30 October and 7 November 2022.* [Personal communication].

Pringle, R. (2019). *The Power of Money: How ideas about money shaped the modern world*. Palgrave Macmillan.

Purcell, O., & Lu, T. K. (2014). Synthetic analog and digital circuits for cellular computation and memory. *Current Opinion in Biotechnology*, *29*, 146–155. https://doi.org/10.1016/j.copbio.2014.04.009

Qiu, J. (2014). Ancient times table hidden in Chinese bamboo strips. *Nature News*. https://doi.org/10.1038/nature.2014.14482

Quirico, O. (2009). *A Purely Formal Theory of Law – The Deontic Network. EUI Working Papers (MWP 2009/11)*. European University Institute Research Repository, Max Weber Programme. http://hdl.handle.net/1814/11482 EUI MWP 2009-11MWP_2009_11.pdf

Qureshi, H. (2017, July 20). *A hacker stole $31M of Ether—How it happened, and what it means for Ethereum*. FreeCodeCamp.Org. https://www.freecodecamp.org/news/a-hacker-stole-31m-of-ether-how-it-happened-and-what-it-means-for-ethereum-9e5dc29e33ce/

Rakočević, M. M. (2018). The Cipher of the Genetic Code. *Biosystems*, *171*, 31–47. https://doi.org/10.1016/j.biosystems.2018.05.009

Rao, S., & Kailath, T. (1984). Orthogonal digital filters for VLSI implementation. *IEEE Transactions on Circuits and Systems*, *31*(11), 933–945. https://doi.org/10.1109/TCS.1984.1085452

Raymond, E. S. (2003). Basics of the Unix Philosophy. In *The Art of Unix Programming*. Addison-Wesley Professional. http://www.catb.org/~esr/writings/taoup/html/ch01s06.html

République Française. (2016). *OpenFisca: Calculer l’impôt, les cotisations, les aides sociales*. Toutes les API. https://api.gouv.fr/les-api/openfisca

Rich, B. (1995). *Clarence Leonard (Kelly) Johnson (1910-1990): A biographical memoir*. National Academy Press. http://www.nasonline.org/publications/biographical-memoirs/memoir-pdfs/johnson-clarence.pdf

Riegler, A. (2005). Editorial. The constructivist challenge. *Constructivist Foundations*, *1*(1), 1–8.

Roark, B., & Sproat, R. W. (2007). *Computational approaches to morphology and syntax*. Oxford University Press.

Robbins, P., McSweeney, K., Waite, T., & Rice, J. (2006). Even Conservation Rules Are Made to Be Broken: Implications for Biodiversity. *Environmental Management*, *37*(2), 162–169. https://doi.org/10.1007/s00267-005-0009-5

Robinson, J. A. (1965). A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the Association for Computing Machinery*, *12*, 23–41.

Robinson, J. A. (1983). Logic programming—Past, present and future. *New Generation Computing*, *1*(2), 107–124. https://doi.org/10.1007/BF03037419

Roermund, B. V. (2013). Rules as Icons: Wittgenstein's Paradox and the Law. *Ratio Juris*, *26*(4), 538–559. https://doi.org/10.1111/raju.12027

Rosa, J. A., Porac, J. F., Runser-Spanjol, J., & Saxon, M. S. (1999). Sociocognitive Dynamics in a Product Market. *Journal of Marketing*, *63*, 64. https://doi.org/10.2307/1252102

Rosenthal, S. B., & Bourgeois, P. L. (1977). Pragmatism, Scientific Method, and the Phenomenological Return to Lived Experience. *Philosophy and Phenomenological Research*, *38*(1), 56. https://doi.org/10.2307/2106514

Roskind, J. (2013). *QUIC: . A design document and specification rationale for multiplexed stream transport over UDP.* Google Docs. https://docs.google.com/document/d/1RNHkx_VvKWyWg6Lr8SZ-saqsQx7rFV-ev2jRFUoVD34/edit?usp=embed_facebook

Ross, R. G. (1997). *Ross, R. (1997). The Business Rule Book: Classifying, Defining, and Modeling Rules (2nd Ed.)*. Business Rule Solutions LLC. https://www.amazon.com/Business-Rule-Book-Classifying-Defining/dp/0941049035

Ross, R. G. (2022, May 10). *Personal communication.* [Personal communication].

Roth, A. (2011). The Necessity of "Necessity": Hume's Psychology of Sophisticated Causal Inference. *Canadian Journal of Philosophy*, *41*(2), 263–287. https://doi.org/10.1353/cjp.2011.0012

Rücker, B. (2020, May 28). *Direct email correspondence about the expression of a sample rule in Decision Modeling Notation (DMN)*. https://camunda.com/about/leadership/

Rushkoff, D. (2010). *Program Or Be Programmed: Ten Commands for a Digital Age*. OR Books.

Russell, A. L. (2006). "Rough Consensus and Running Code" and the Internet-OSI Standards War. *IEEE Annals of the History of Computing*, *28*(3), 48–61. https://doi.org/10.1109/MAHC.2006.42

Russell, B. (1919). The Philosophy of Logical Atomism: Lectures 5-6, The Hegeler Institute. *Monist*, *29*(2), 190–222. https://doi.org/10.5840/monist19192922

SAE. (2018). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles J3016_201806*. SAE International. https://www.sae.org/standards/content/j3016_201806/

Saint-Exupéry, A. de. (1939). *Wind, Sand and Stars*. Houghton Mifflin Harcourt.

Sampson, S. E. (2015). *Essentials of Service Design and Innovation: Developing High-value Service Businesses with PCN Analysis*. SES (Scott E. Sampson.

Sandiford, S. (2021, December 6). *Xalgorithms 5th Anniversary (2016-2021)* [MP4]. Animated video created by Sciconic Science Media (https://sciconic.com) for Xalgorithms Foundation, for presentation at the "Solutions Showcase" of the US Federal Reserve's "Faster Payments Task Force" (FPTF). https://vimeo.com/653873375

Santanu Mandal. (2014). Supply chain resilience: A state-of-the-art review and research directions. *International Journal of Disaster Resilience in the Built Environment*, *5*(4), 427–453. https://doi.org/10.1108/IJDRBE-03-2013-0003

Savchuk, M. M., & Fesenko, A. V. (2019). Quantum Computing: Survey and Analysis. *Cybernetics and Systems Analysis*, *55*(1), 10–21. https://doi.org/10.1007/s10559-019-00107-w

Schartum, D. W. (2016a). *From Algorithmic Law to Automation-friendly Legislation*. http://sclbc.zehuti.co.uk/site.aspx?i=ed48414

Schartum, D. W. (2016b). Law and algorithms in the public domain. *Etikk i Praksis - Nordic Journal of Applied Ethics*, *10*(1), 15. https://doi.org/10.5324/eip.v10i1.1973

Schein, E. (1995). *Learning Consortia: How to create Parallel Learning Systems for Organization Sets. Working Paper, Organizational Learning Center.* MIT Sloan School of Management, Cambridge, Mass.

Scherer, S. (2020, October 19). Canada and U.S. border to stay closed to non-essential travel until Nov. 21. *Financial Post*. https://financialpost.com/pmn/business-pmn/canada-and-u-s-border-to-stay-closed-to-non-essential-travel-until-nov-21-2

Schmandt-Besserat, D. (2009). Tokens and Writing: The Cognitive Development. *Scripta*, *1*(1), 145–154.

Schneider, M. (2018). *OpenFisca Documentation*. https://openfisca.org/doc/

Schneider, N. (2015). What I've learned about annotating informal text (and why you shouldn't take my word for it). *Proceedings of The 9th Linguistic Annotation Workshop*, 152–157. https://doi.org/10.3115/v1/W15-1618

Schumpeter, J. A., & Boody, E. (1954). *History of Economic Analysis*. George Allen and Unwin.

Schumpter, J. (1942). *Capitalism, Socialism, and Democracy*. Harper.

Sedig, K., Rowhani, S., & Liang, H.-N. (2005). Designing interfaces that support formation of cognitive maps of transitional processes: An empirical study. *Interacting with Computers*, *17*(4), 419–452. https://doi.org/10.1016/j.intcom.2005.02.002

Sergot, M. J., Sadri, F., Kowalski, R. A., Kriwaczek, F., Hammond, P., & Cory, H. T. (1986). The British Nationality Act as a logic program. *Communications of the ACM*, *29*(5), 370–386. https://doi.org/10.1145/5689.5920

Severance, C. (2012). JavaScript: Designing a Language in 10 Days. *IEEE Computer*, *45*(2), 7–8. https://doi.org/10.1109/MC.2012.57

Shafer, G. (1996). The significance of Jacob Bernoulli's Ars Conjectandi for the philosophy of probability today. *Journal of Econometrics*, *75*(1), 15–32. https://doi.org/10.1016/0304-4076(95)01766-6

Shannon, C. (1948). A Mathematical Theory of Communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, *5*(1), 3–55.

Shannon, C. E. (1998). Communication in the Presence of Noise. *Proceedings of the IEEE*, *86*(2), 11.

Shental, O., & Kanter, I. (2009). *The second law of thermodynamics for communication channels*. 784–788. https://doi.org/10.1109/EEEI.2008.4736643

Shindler, J. (2010). *Transformative classroom management: Positive strategies to engage all students and promote a psychology of success* (1st ed). Jossey-Bass.

Shosky, J. (1997). Russell's Use of Truth Tables. *Russell: The Journal of Bertrand Russell Studies*, *17*(1). https://doi.org/10.15173/russell.v17i1.1912

Simon, W. H. (2009). *The Practice of Justice: A theory of lawyers' ethics*. Harvard University Press.

Simons, J. (2012). *Don't be afraid of vanity projects*. Association for Project Management. http://www.apm.org.uk/blog/don%E2%80%99t-be-afraid-vanity-projects#.UyX-vdHNV0w

Simpson, B., & den Hond, F. (2022). The Contemporary Resonances of Classical Pragmatism for Studying Organization and Organizing. *Organization Studies*, *43*(1), 127–146. https://doi.org/10.1177/0170840621991689

Singh, V. (2014). Recent advances and opportunities in synthetic logic gates engineering in living cells. *Systems and Synthetic Biology*, *8*(4), 271–282. https://doi.org/10.1007/s11693-014-9154-6

Sinkovics, R., & Alfoldi, E. (2012). Progressive Focusing and Trustworthiness in Qualitative Research: The Enabling Role of Computer-Assisted Qualitative Data Analysis Software (CAQDAS). *Management International Review*, *52*(6), 817–845. https://doi.org/10.1007/s11575-012-0140-5

Sirakoulis, G. Ch., Karafyllidis, I., Sandaltzopoulos, R., Tsalides, Ph., & Thanailakis, A. (2004). An algorithm for the study of DNA sequence evolution based on the genetic code. *Biosystems*, *77*(1–3), 11–23. https://doi.org/10.1016/j.biosystems.2004.02.006

Siuti, P., Yazbek, J., & Lu, T. K. (2013). Synthetic circuits integrating logic and memory in living cells. *Nature Biotechnology*, *31*(5), 448–452. https://doi.org/10.1038/nbt.2510

Skiena, S. (2008). *The Algorithm Design Manual* (Vol. 2). Springer-Verlag.

Skuce, D. (1977). *Towards Communicating Qualitative Knowledge Between Scientists and Machines* [PhD (Electrical Engineering), McGill University]. https://escholarship.mcgill.ca/concern/theses/ns064669x

Skuce, D. (2003). *A Controlled Language for Knowledge Formulation on the Semantic Web*. Academia.edu. https://www.academia.edu/75271661/A_Controlled_Language_for_Knowledge_Formulation_on_the_Semantic_Web

Sliwa, C., & King, J. (2000). B-to-B hard to spell with XML. *Computerworld*, *34*(9), 1,97.

Smiley, T. (1960). Sense without Denotation. *Analysis*, *20*(6), 125–135. https://doi.org/10.2307/3326888

Smith, A. J. (1987). *Design of CPU Cache Memories*. IEEE Region 10 Conference (TENCON), Seoul, South Korea. https://www2.eecs.berkeley.edu/Pubs/TechRpts/1987/CSD-87-357.pdf

Smith, M. G. (1960). Social and cultural pluralism. In V. Rubin (Ed.), *Social and Cultural Pluralism in the Caribbean* (Vol. 83, pp. 763–785).

Smith, P., Hutchison, D., Sterbenz, J. P. G., Schöller, M., Fessi, A., Karaliopoulos, M., Lac, C., & Plattner, B. (2011). Network resilience: A systematic approach. *IEEE Communications Magazine*, *49*(7), 88–97. https://doi.org/10.1109/MCOM.2011.5936160

Söderlund, J. (2004). On the broadening scope of the research on projects: A review and a model for analysis. *International Journal of Project Managemen*, *22*(8), 655–667.

Sousa, T., Brockway, P. E., Cullen, J. M., Henriques, S. T., Miller, J., Serrenho, A. C., & Domingos, T. (2017). The Need for Robust, Consistent Methods in Societal Exergy Accounting. *Ecological Economics*, *141*, 11–21. https://doi.org/10.1016/j.ecolecon.2017.05.020

Spaccasassi, C., Lakin, M. R., & Phillips, A. (2019). A Logic Programming Language for

Computational Nucleic Acid Devices. *ACS Synthetic Biology*, *8*(7), 1530–1547. https://doi.org/10.1021/acssynbio.8b00229

Spencer, H. (1898). *The Principles of Sociology* (Vol. 1–4). D. Appleton & Company.

Spielthenner, G. (2017). The Is-Ought Problem in Practical Ethics. *HEC Forum*, *29*(4), 277–292. https://doi.org/10.1007/s10730-016-9318-8

Sporny, M., & Potvin, J. (2014). *Submission to a Public Consultation: US Federal Reserve Payment Improvements – Web Payments Response*. US Federal Reserve. https://fedpaymentsimprovement.org/wp-content/uploads/Manu_Sporny-World_Wide_Web_Consortium-121113.pdf

Srivastava, J., & Shu, L. H. (2014). The Affordance of Absence. *Volume 7: 2nd Biennial International Conference on Dynamics for Design; 26th International Conference on Design Theory and Methodology*, V007T07A029. https://doi.org/10.1115/DETC2014-35285

Staal, F. (1975). *Exploring Mysticism: A Methodological Essay*. University of California Press.

Stallman, R. (1991). *Why Software Should Be Free.* Free Software Foundation. http://www.gnu.org/philosophy/shouldbefree.html

Stallman, R. M. (1991). *Why Software Should Be Free*. http://www.gnu.org/philosophy/shouldbefree.html

Stalnaker, S. (2016). *Hub Culture and Partners Propose Architecture for Faster Payments*. Hub Culture. https://hubculture.com/hubs/801/news/746/

Stalnaker, S., Murray, W., Johnson, J., & Potvin, J. (2016). *RAIN and RAIL (Real-time Asset Interchange Network on a Real-time Asset Interchange Ledger)*. Submission to the “Qualified Independent Assessment Team” (QIAT), US Federal Reserve’s “Faster Payments Task Force” (FPTF) convened under contract by McKinsey & Company. https://fasterpaymentstaskforce.org/wp-content/uploads/hub-culture-vs.pdf

Steinbuch, K. (1957). *Informatik, Automatische Informationsverarbeitung*. SEG-Nachrichten. Technische Mitteilungen Der Standard Elektrik Gruppe.

Steinbuch, K. (1958). Informatik. *Zeitschrift Für Angewandte Mathematik Und Mechanik*, *38*, 319–330.

Steiner, E. (1998). *Methodology of Theory Building*. Educology Research Associates.

Sterbenz, J. P. G., Hutchison, D., Çetinkaya, E. K., Jabbar, A., Rohrer, J. P., Schöller, M., & Smith, P. (2010). Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, *54*(8), 1245–1265. https://doi.org/10.1016/j.comnet.2010.03.005

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F., & Balakrishnan, H. (2003). Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, *11*(1), 17–32. https://doi.org/10.1109/TNET.2002.808407

Strack, F., & Deutsch, R. (2004). Reflective and Impulsive Determinants of Social Behavior. *Personality and Social Psychology Review*, *8*(3), 220–247. https://doi.org/doi:10.1207/s15327957pspr0803_1

Strauch, B. (2018). Ironies of Automation: Still Unresolved After All These Years. *IEEE Transactions on Human-Machine Systems*, *48*(5), 419–433. https://doi.org/10.1109/THMS.2017.2732506

Stucky, B. (2020). *Rules, Decisions and... Standards?* BPMInstitute.Org. https://www.bpminstitute.org/resources/articles/rules-decisions-and-standards

Studd, J. (2019). *Everything, more or less: A defence of generality relativism*. Oxford University Press.

Sturm, S. P. (2019). Lawyering Paradoxes: Making Meaning of the Contradictions. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3477424

Sullivan, L. H. (1896). The Tall Office Building Artistically Considered. *Lippincotts Magazine*, 7.

Suprapto, M., Bakker, H. L. M., Mooi, H. G., & Hertogh, M. J. C. M. (2015). How do contract types and incentives matter to project performance? *International Journal of Project Management*. https://doi.org/10.1016/j.ijproman.2015.08.003

Swedlow, J. R., Zanetti, G., & Best, C. (2011). Channeling the data deluge. *Nature Methods*, *8*(6), 3.

Syed, J., Mingers, J., & Murray, P. A. (2010). Beyond rigour and relevance: A critical realist approach to business education. *Management Learning*, *41*(1), 71–85. https://doi.org/10.1177/1350507609350839

Taylor, R. B. (1951). *Musson's Improved Ready Reckoner: Form and Log Book (Revised and Enlarged)*. Musson Book Company.

TBS. (2020). *Agile Nations Charter*. Treasury Board of Canada Secretariat. https://www.canada.ca/en/government/system/laws/developing-improving-federal-regulations/modernizing-regulations/agile-nations-charter.html

Tekic, Z., & Koroteev, D. (2019). From disruptively digital to proudly analog: A holistic typology of digital transformation strategies. *Business Horizons*, *62*(6), 683–693. https://doi.org/10.1016/j.bushor.2019.07.002

Temnikov, A., & Podshivalova, M. (2022). Digital Transformation of Industry: Benefits, Costs and Risks (Original title: ЦИФРОВАЯ ТРАНСФОРМАЦИЯ ПРОМЫШЛЕННОСТИ: ВЫГОДЫ, ЗАТРАТЫ И РИСКИ) (Deep-L Translator, Trans.). *Bulletin of SUSU. Economics and Management Series.*, *16*(2), 122–131. https://doi.org/10.14529/em220212

Tencent. (2019). *Experimental Security Research of Tesla Autopilot*. Tencent Keen Security Lab. https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf

The Economist Group. (2013, September 23). *EDGE-NYC 2013 Conference*. (EdgeNYC2013), Google New York City. https://edgeconf.com/2013-nyc

Thomé, A. M. T., Scavarda, L. F., Scavarda, A., & Thomé, F. E. S. de S. (2016). Similarities and contrasts of complexity, uncertainty, risks, and resilience in supply chains and temporary multi-organization projects. *International Journal of Project Management*, *34*(7), 1328–1346. https://doi.org/10.1016/j.ijproman.2015.10.012

Thompson, N. A., & Byrne, O. (2022). Imagining Futures: Theorizing the Practical Knowledge of Future-making. *Organization Studies*, *43*(2), 247–268. https://doi.org/10.1177/01708406211053222

Thomson, M. (2021). *Application-Layer Protocol Negotiation (ALPN) for WebRTC. RFC 8833* [Request for Comments]. Internet Engineering Task Force. https://doi.org/10.17487/RFC8833

Thoreau, H. D. (2008). *Civil Disobedience: Resistance to Civil Government*. The Floating Press. https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=313566&lang=fr&site=ehost-live

Thornhill, C. (2012). National Sovereignty and the Constitution of Transnational Law: A Sociological Approach to a Classical Antinomy. *Transnational Legal Theory*, *3*(4), 394–460. https://doi.org/10.5235/20414005.3.4.394

TransportXtra. (2019, February 4). *Rail-Powered Property - Property-Powered Rail: A transformational approach?* https://www.transportxtra.com/publications/local-transport-today/news/60202/rail-powered-property--property-powered-rail-a-transformational-approach-/

Treisman, A. M., & Gelade, G. (1980). A Feature-Integration Theory of Attention. *Cognitive Psychology*, *12*, 97–136.

Tribus, M., Shannon, P. T., & Evans, R. B. (1966). Why thermodynamics is a logical consequence of information theory. *AIChE Journal*, *12*(2), 244–248. https://doi.org/10.1002/aic.690120208

Trypuz, R., & Kulicki, P. (2015). Jerzy Kalinowski's Logic of Normative Sentences Revisited. *Studia Logica*, *103*(2), 389–412. https://doi.org/10.1007/s11225-014-9572-1

Tukamuhabwa, B. R., Stevenson, M., Busby, J., & Zorzini, M. (2015). Supply chain resilience: Definition, review and theoretical foundations for further study. *International Journal of Production Research*, *53*(18), 5592–5623. https://doi.org/10.1080/00207543.2015.1037934

Tukey, J. W. (1962). The Future of Data Analysis. *The Annals of Mathematical Statistics*, *33*(1), 1–67.

Turing, A. M. (1937). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, *s2-42*(1), 230–265. https://doi.org/10.1112/plms/s2-42.1.230

Ulanowicz, R. E. (2001). Information theory in ecology. *Computers & Chemistry*, *25*(4), 393–399. https://doi.org/10.1016/S0097-8485(01)00073-0

UNCTAD. (2016, July 1). UNCTAD 14. *Proceedings of UNCTAD 14*. https://unctad.org/search?keys=UNCTAD+14+proceedings

Underwood, W. (2011). *Tuple Oriented Programming*. http://wiki.c2.com/?TupleOrientedProgramming

UNDP. (2018). *United Nations Standard Products and Services Code (UNSPSC). Managed by GS1 US for the UN Development Programme (UNDP).* https://www.unspsc.org/

UNSD. (2018). *International Standard Industrial Classification of All Economic Activities, Rev.4)*. United Nations Statistics Division. http://unstats.un.org/unsd/cr/registry/regcst.asp?Cl=27

UQO. (2011). *Un proposition de programme académique: Doctorat en administration—Gestion de projet*. Université du Québec en Outaouais.

U.S. Senate. (1814). *Treaty of Ghent*. https://www.senate.gov/about/powers-procedures/treaties/treaty-of-ghent/transcript-treaty-of-ghent.htm

Vaishnavi, V. K., & Kuechler, W. (2015). *Design Science Research Methods and Patterns: Innovating Information and Communication Technology, 2nd Edition* (2 edition). CRC Press.

Van De Ven, A. H., & Johnson, P. E. (2006). Knowledge for Theory and Practice. *Academy of Management Review*, *31*(4), 802–821. https://doi.org/10.5465/amr.2006.22527385

van der Haghen, H. (2016). *Tetranary Logic for Self-Programming Software or Hardware*.

http://logic.zien.info/

van Dongen, K., & van Maanen, P.-P. (2013). A framework for explaining reliance on decision aids. *International Journal of Human-Computer Studies, 71*(4), 410–424. https://doi.org/10.1016/j.ijhcs.2012.10.018

Van Fraassen, B. C. (1980). *The scientific image*. Clarendon Press ; Oxford University Press.

Van Woensel., W. (2022, April 11). *Direct email correspondence about the expression of a sample rule in Notation3* [Personal communication].

Vanthienen, J. (2010). Rules as Data: Decision Tables and Relational Databases. *Business Rules Journal*, *11*(1). https://www.brcommunity.com/articles.php?id=b516

Vanthienen, J. (2012a). *The History of Modeling Decisions using Tables (Part 1): Commentary. Business Rules Journal, Vol. 13, No. 2,*. Business Rules Community - BRC. https://www.brcommunity.com/articles.php?id=b637

Vanthienen, J. (2012b). *The History of Modeling Decisions using Tables (Part 2): Commentary . Business Rules Journal, Vol. 13, No. 3*. Business Rules Community - BRC. https://www.brcommunity.com/articles.php?id=b641

Vanthienen, J. (2012c). *The History of Modeling Decisions using Tables (Part 3): Commentary : Business Rules Community / Business Rules Journal*. Business Rules Community - BRC. https://www.brcommunity.com/articles.php?id=b652

Vanthienen, J. (2022, April 13). *Direct video call about logic gates and decision tables* [Personal communication].

Vanthienen, J., & Dries, E. (1992). *Developments in Decision Tables: Evolution, Applications and a Proposed Standard. ONDERZOEKSRAPPORT NR 9227*. Katholieke Universiteit Leuven Department of Applied Economic Sciences. https://lirias.kuleuven.be/retrieve/274168

Vanthienen, J., & Dries, E. (1993). *A New Approach to the Use of Decision Tables in Software Engineering. DTEW Research Report 9310.* Faculty of Economics and Business KU Leuven. https://lirias.kuleuven.be/retrieve/274627

Vardi, M. Y. (2012). What is an algorithm? *Communications of the ACM*, *55*(3), 5–5. https://doi.org/10.1145/2093548.2093549

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762). arXiv. http://arxiv.org/abs/1706.03762

Venn, J. (1880). On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *10*(59), 1–18. https://doi.org/10.1080/14786448008626877

Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Qi Dong, J., Fabian, N., & Haenlein, M. (2021). Digital transformation: A multidisciplinary reflection and research agenda. *Journal of Business Research*, *122*, 889–901. https://doi.org/10.1016/j.jbusres.2019.09.022

Verhulst, A., Depraetere, I., & Heyvaert, L. (2013). Source and strength of modality: An empirical study of root should, ought to and be supposed to in Present-day British English. *Journal of Pragmatics*, *55*, 210–225. https://doi.org/10.1016/j.pragma.2013.05.010

von Landesberger, T., Fellner, D. W., & Ruddle, R. A. (2017). Visualization System Requirements for Data Processing Pipeline Design and Optimization. *IEEE Transactions on Visualization and Computer Graphics*, *23*(8), 2028–2041. https://doi.org/10.1109/TVCG.2016.2603178

Von Wright, G. (1999). Deontic logic—As I see it. Paper presented at the Fourth International Workshop on Deontic Logic in Computer Science (DEON'98), Bologna, 1998. In P. McNamara & H. Prakken (Eds.), *Norms, Logics and Information Systems: New Studies in Deontic Logic and Computer Science* (pp. 15–28). IOS Press.

Von Wright, G. H. (1951). Deontic Logic. *Mind*, *60*(237), 1–15.

Von Wright, G. H. (1999). Deontic Logic: A Personal View. *Ratio Juris*, *12*(1), 26–38. https://doi.org/10.1111/1467-9337.00106

Von Wright, G. H. (2000). On Norms and Norm Propositions: A Sketch. In W. Krawietz, R. S. Summers, & O. Weinberger (Eds.), *The Reasonable as Rational?* (pp. 173–178). Duncker & Humblot.

W3C. (2014). Web & Payments: How do you want to pay? *W3C Workshop, 24-25 March 2014*. http://www.w3.org/2013/10/payments/agenda.html

W3C. (2019). *CSS Writing Modes Level 3. CSS Working Group Recommendation*. https://www.w3.org/TR/css-writing-modes-3/

WEF. (2022). *Regulatory Technology for the 21st Century. White Paper*. World Economic Forum, In collaboration with the Global Futures Council on Agile Governance. https://www3.weforum.org/docs/WEF_Regulatory_Tech_for_the_21st_Century_2022.pdf

Wei, Z., Fu, W., Liu, Q., Jing, H., Jin, C., Chen, Y., Xia, W., Zhu, X., & Xu, D. (2019). Construction of Boolean logic gates based on dual-vector circuits of multiple gene regulatory elements. *Molecular Genetics and Genomics*, *294*(2), 277–286. https://doi.org/10.1007/s00438-018-1502-x

Weissman, D. (2006). *The Cage: Must, Should, and Ought From Is*. State University of New York Press. https://proxybiblio.uqo.ca:2058/ehost/ebookviewer/ebook/bmxlYmtfXzE3MTk4OV9fQU41?sid=129a6e30-d1b6-4101-8bc7-a79d53371d26@sdc-v-sessmgr01&vid=0&format=EB&rid=1

Weitzner, D. J., Hendler, J., Berners-Lee, T., & Connolly, D. (2006). Creating a Policy-Aware Web: Discretionary, Rule-based Access for the World Wide Web. In E. Ferrari & B. Thuraisingham (Eds.), *Web and Information Security* (pp. 1–31). IGI Global. http://doi:10.4018/978-1-59140-588-7.ch001

Wenzel, M. (2022). Taking the Future More Seriously: From Corporate Foresight to "Future-Making." *Academy of Management Perspectives*, *36*(2), 845–850. https://doi.org/10.5465/amp.2020.0126

Wenzel, M., Krämer, H., Koch, J., & Reckwitz, A. (2020). Future and Organization Studies: On the rediscovery of a problematic temporal category in organizations. *Organization Studies*, *41*(10), 1441–1455. https://doi.org/10.1177/0170840620912977

Whitehouse, M. (2005). Slices of risk: How a formula ignited a market that burned some big investors. Reproduced online: Http://math.bu.edu/people/murad/MarkWhitehouseSlicesofRisk.txt. *The Wall Street Journal*, *September 12, 2005*.

Whyte, J. (2019). How Digital Information Transforms Project Delivery Models. *Project Management Journal*, *50*(2), 177–194. https://doi.org/10.1177/8756972818823304

Whyte, J., Comi, A., & Mosca, L. (2022). Making futures that matter: Future making, online working and organizing remotely. *Organization Theory*, *3*(1), 263178772110691.

https://doi.org/10.1177/26317877211069138

Whyte, J., & Mottee, L. (2022). Projects as interventions. *International Journal of Project Management*, *40*(8), 934–940. https://doi.org/10.1016/j.ijproman.2022.10.007

Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer.

Wiley. (2022). *Regulation & Governance*. Wiley Online Library. https://doi.org/10.1111/(ISSN)1748-5991

Wintner, S., & Sarkar, A. (2002). A Note on Typing Feature Structures. *Computational Linguistics*, *28*(3), 389–397. https://doi.org/10.1162/089120102760276027

Wirth, N. (1976). *Algorithms + Data Structures = Programs*. Prentice-Hall.

Wittgenstein, L. (1953). *Philosophical Investigations: The German Text, with a Revised English Translation. 50th Anniversary Edition (1991)* (3rd ed.). Wiley-Blackwell.

Wittgenstein, L. (1991). *Philosophical Investigations: The German Text, with a Revised English Translation. 50th Anniversary Edition* (3rd ed.). Wiley-Blackwell.

Woolley, J. B., & Stone, N. D. (1987). Application of Artificial Intelligence to Systematics: Systex-A Prototype Expert System for Species Identification. *Systematic Zoology*, *36*(3), 248. https://doi.org/10.2307/2413065

World Bank. (2019). *Data-Driven Development. Information and Communications for Development*. The World Bank. 10.1596/978-1-4648-1325-2

Wright, A. S. (2016). The Physics of Forgetting: Thermodynamics of Information at IBM 1959–1982. *Perspectives on Science*, *24*(1), 112–141. https://doi.org/10.1162/POSC_a_00194

Wu, T. (2003). Network Neutrality, Broadband Discrimination. *Journal on Telecom and High Tech Law*, *2*, 141–180.

Wu, T. (2022). Digital project management: Rapid changes define new working environments. *Journal of Business Strategy*, *43*(5), 323–331. https://doi.org/10.1108/JBS-03-2021-0047

Xalgorithms. (2016, July 7). *Introduction to Xalgorithms*. https://vimeo.com/173809271

Xalgorithms. (2021, December 6). *Xalgorithms 5th Anniversary (2016-2021)*. https://vimeo.com/653873375

(2019) (testimony of Xalgorithms Foundation). https://meet.jit.si/xalgorithms

Xalgorithms Foundation. (2021, 2022). *Specifications and Reference Implementation Components for an Internet of Rules (IoR)*. GitLab. https://gitlab.com/xalgorithms-alliance

Xie, Z., Liu, S. J., Bleris, L., & Benenson, Y. (2010). Logic integration of mRNA signals by an RNAi-based molecular computer. *Nucleic Acids Research*, *38*(8), 2692–2701. https://doi.org/10.1093/nar/gkq117

Yanofsky, N. S. (2003). A Universal Approach to Self-Referential Paradoxes, Incompleteness and Fixed Points. *Bulletin of Symbolic Logic*, *9*(3), 362–386. https://doi.org/10.2178/bsl/1058448677

Yates, F. E. (1985). Bioelectronic and Molecular Electronic Devices. *Proc. Int. Symp. on Future Electron Devices*.

Yusuf, M. (2022, May 27). Datasette—Simon Willison. *Architecture Notes*. https://architecturenotes.co/datasette-simon-willison/

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353.

https://doi.org/10.1016/S0019-9958(65)90241-X

Zahra, S. A., & George, G. (2002). Absorptive Capacity: A review, reconceptualization, and extension. *Academy of Management Review*, *27*(2), 185–203.

Zdankus, J., & Delli Colli, A. (2021). *Getting the most from your data-driven transformation: 10 key principles*. MIT Technology Review. https://www.technologyreview.com/2021/10/14/1037054/getting-the-most-from-your-data-driven-transformation-10-key-principles/

Zeng, J., & Plale, B. (2013). Data Pipeline in MapReduce. *2013 IEEE 9th International Conference on E-Science*, 164–171. https://doi.org/10.1109/eScience.2013.21

Zhao, Y., & Chakrabarty, K. (2010). Digital Microfluidic Logic Gates and Their Application to Built-in Self-Test of Lab-on-Chip. *IEEE Transactions on Biomedical Circuits and Systems*, *4*(4), 250–262. https://doi.org/10.1109/TBCAS.2010.2048567

Zhou, K. Z., & Poppo, L. (2010). Exchange hazards, relational reliability, and contracts in China: The contingent role of legal enforceability. *Journal of International Business Studies*, *41*(5), 861–881. https://doi.org/10.1057/jibs.2010.7