UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

DOCTORAL THESIS
PRESENTED TO
DÉPARTEMENT DE SCIENCES ET TECHNOLOGIES DE L'INFORMATION

IN PARTIAL FULFILLEMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

BY
MILOUD ELOUMRI

TOWARDS EFFICIENT ERLANG-BASED JOB-APPLICANT MATCHING: INTEGRATING SERESYE, SEMANTIC WEB TOOLKIT, AND MCDM

GATINEAU, JULY 8, 2024

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

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

Mr. José Bélanger, Member of the jury
Département des sciences administratives, Université du Québec en Outaouais

Mrs. Alvine Boaye Belle, External Evaluator
Department of Electrical Engineering and Computer Science, York University

Mrs. Rokia Missaoui, Internal Evaluator
Département d'informatique et d'ingénierie, Université du Québec en Outaouais

2024

AT UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Stephane Gagnon. His endless efforts, constant support, and thorough guidance have been vital in making this thesis possible.

My friends have been a constant source of support and inspiration. I will always be grateful to them.

I am extremely grateful to my family for their endless patience, inspiring motivation, and constant support. I am particularly thankful to my cousin, Mohamed Ramadan Eloumri, who continues to inspire me, despite his absence.

Last but not least, I dedicate this thesis and all my work to my late parents, Salem, and Salma Eloumri. Their love, compassion, support and hopes for me are embodied in each word in my thesis. They have been my guiding light. To them, I owe everything.

SOMMAIRE

La Gestion des Ressources Humaines (GRH) est cruciale pour le succès organisationnel. Elle implique la gestion de plusieurs processus clés, y compris le recrutement et la sélection. Le processus de recrutement se concentre sur l'attraction de candidats appropriés, tandis que le processus de sélection implique de choisir les meilleurs candidats en fonction des besoins de l'organisation. Ces processus contribuent significativement à obtenir des avantages concurrentiels, mais ils font face à plusieurs défis pouvant augmenter les coûts et compliquer la tâche de sélection du meilleur candidat.

Cette thèse utilise le langage de programmation fonctionnel Erlang pour améliorer l'efficacité de la sélection des meilleurs candidats pour un poste donné. Erlang est connu pour sa scalabilité, sa concurrence, sa fiabilité, son traitement en temps réel et ses capacités de correspondance de modèles, lui permettant de gérer efficacement de grands ensembles de données complexes. Cependant, l'application d'Erlang dans la sélection du personnel est largement inexplorée. Par conséquent, cette thèse vise à développer un système sophistiqué de correspondance emploi-candidat intégrant les capacités de moteur de règles de SERESYE (Swarm-oriented ERlang Expert System Engine), la fonctionnalité de modélisation de données web sémantiques du Semantic Web Toolkit pour les applications Erlang, et les principes de la Prise de Décision Multi-Critères (MCDM) utilisant la Méthode de Somme Pondérée (WSM) pour permettre un traitement complet des règles et une évaluation multi-critères des qualifications des candidats par rapport aux exigences du poste. De plus, la thèse explore l'élargissement du champ d'interprétation des données du système de correspondance en utilisant RDFLib, une bibliothèque Python pour l'analyse de la plupart des formats de données sémantiques.

L'étude met en œuvre un cas d'utilisation prototype dans le domaine des parentés pour explorer le concept de correspondance de règles sur les données du web sémantique en utilisant les technologies Erlang mentionnées. Cependant, la recherche applique principalement l'approche de correspondance à un domaine complexe représenté par les données d'emplois et de candidats en Gestion de la Technologie des Affaires (BTM) modélisées à l'aide de l'éditeur d'ontologies Protégé.

Les résultats démontrent la faisabilité d'intégrer le traitement des règles SERESYE avec des ensembles de données du web sémantique, conduisant à une amélioration de la correspondance emploi-candidat. Cette thèse apporte ainsi une contribution significative à la sélection du personnel, aux systèmes experts, au web sémantique et aux systèmes de prise de décision, fournissant une base pour les avancées futures.

Les travaux futurs comprennent l'adressage des limitations de la mise en œuvre actuelle. Le développement d'une interface web pour le système de correspondance emploi-candidat en utilisant le cadre Zotonic d'Erlang. L'utilisation de plusieurs moteurs de règles parallèles SERESYE et la bibliothèque Poolboy d'Erlang pour créer et gérer un pool de processus concurrents. Les extensions d'application pour inclure des opérations connexes telles que l'envoi d'e-mails, la planification d'entretiens et la formulation d'offres d'emploi.

Mots-clés: Sélection du Personnel, Correspondance emploi-candidat, Erlang, SERESYE, Toolkit Web Sémantique pour les applications Erlang, Prise de Décision Multi-Critères (MCDM), Méthode de Somme Pondérée (WSM), RDFLib, Technologies du Web Sémantique, Emplois en Gestion de la Technologie des Affaires (BTM).

ABSTRACT

Human Resources Management (HRM) is crucial for organizational success. It involves managing several key processes, including recruitment and selection. The recruitment process focuses on attracting suitable applicants, while the selection process involves choosing the best candidates based on organizational needs. These processes significantly contribute to gaining competitive advantages, but they face several challenges that can increase costs and complicate the task of selecting the best candidate.

This thesis uses Erlang functional programming language to enhance the efficiency of selecting the best applicants for a certain position. Erlang is known for its scalability, concurrency, reliability, real-time processing, and pattern matching capabilities, enabling it to effectively handle large and complex datasets. However, Erlang's application in personnel selection is largely unexplored. Therefore, the thesis aims to develop a sophisticated job-applicant matching and evaluation application that integrates the rule engine capabilities of SERESYE (Swarm-oriented ERlang Expert System Engine), the semantic wed data modeling functionality of Semantic Web Toolkit for Erlang Applications, and the principles of Multi-Criteria Decision Making (MCDM) using Weighted Sum Method (WSM) to enable comprehensive rule processing and multi-criteria evaluation of applicants qualifications against job requirements. Additionally, the thesis explores expanding the matching system's data interpretation scope using RDFLib, a Python library for parsing most semantic data syntaxes.

The study implements a prototype use case represented by a simple ontology in the domain of relatives to explore the concept of rule matching over semantic web data using the mentioned Erlang technologies. However, the research primarily applies the matching approach to a complex area represented by Business Technology Management (BTM) jobs and applicants ontology modeled using Protégé ontology editor.

The findings demonstrate the feasibility of integrating SERESYE rule processing with semantic web datasets, leading to enhanced job-applicant matching. This thesis thus makes a significant contribution to personnel selection, expert systems, semantic web, and decision-making systems, providing a foundation for future advancements.

Future work includes addressing the limitations of current implementation. The development of a web interface for the job-applicant matching system using Erlang's Zotonic framework. The use of SERESYE multiple parallel rule engines and Erlang Poolboy library for creating and managing a pool of concurrent process. The application extensions to include related operations such as emailing, scheduling interviews, and making job offers.

Keywords: Personnel Selection, Jobs applicants matching, Erlang, ERESYE , SERESYE, Semantic Web Toolkit for Erlang Applications, Multi-Criteria Decision Making (MCDM), Weighted Sum Method (WSM), RDFLib, Semantic Web, Ontology, Business Technology Management (BTM) jobs.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
ACL	Agent Communication Language
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
AIRM	Aggregated Indices Randomization Method
ANNs	Artificial Neural Networks
ANP	Analytic Network Process
API	Application Programming Interface
ARAS	Additive Ratio Assessment
ARGUS	Automatic Routine Generating and Updating System
BIBs	Biographical Information Blanks
BTM	Business Technology Management
BWM	Best-Worst Method
BWM	Best Worst Method
CBR	Case-Based Reasoning
CERN	The European Organization for Nuclear Research
CGSC	Command and General Staff College
CLIPS	C-Language Integrated Production System
CoCoSo	COmbined COmpromise SOlution
COMET	Characteristic Objects Method
COPRAS	Complex Proportional Assessment
CRITIC	Criteria Importance Through Intercriteria Correlation
CSV	Comma-Separated Values
DAAs	Data Analytics Algorithms
DAI	Distributed Artificial Intelligence
DANP	DEMATEL-based Analytic Network Process
DEA	Data Envelopment Analysis
DEMATEL	Decision Making Trial and Evaluation Laboratory
DeXi	Dual Expert Interactive
DL	Description Logic
DRSA	Dominance-based Rough Set Approach
DS	Design Science
DSRM	Design Science Research Methodology
EDAS	Evaluation based on Distance from Average Solution
eJason	Erlang Jason
ELECTRE	ÉLimination Et Choix Traduisant la REalité ("Elimination and Choice Translating Reality")
EMAS	Evolutionary Multi-Agent Systems
ERESYE	ERlang Expert SYstem Engine
ERTS	Erlang Runtime System
EVAMIX	Evaluation of Mixed Data

Abbreviation	Meaning
eXAT	erlang eXperimental Agent Tool
FIPA	Foundation for Intelligent Physical Agents
FOL	First Order Logic
FSM	Finite State Machines
FST	Fuzzy Set Theory
FUCOM	Full Consistency Method
GA	Genetic Algorithm
GAIA	Geometrical Analysis for Interactive Aid
GDM	Group Decision-Making
GP	Goal Programming
GRA/GRM	Grey Relational Analysis/Grey Relational Model
HR	Human Resources
HRM	Human Resources Management
HRMES	Human Resource Management Expert System
IDOCRIW	Integrated Determination of Objective Criteria Weights
IDRA	Intercriteria Decision Rule Approach
IFNs	Intuitionistic Fuzzy Numbers
IOWA	Induced Ordered Weighted Averaging
IRI	Internationalized Resource Identifier
IS	Information Systems
IT	Information Technology
ITAC	Information Technology Association of Canada
Jason	Java-based interpreter for an extended version of AgentSpeak language
JESS	Java Expert System Shell
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
KANO	KANO model
KB	Knowledge Base
LAN	Local Area Network
LBWA	Level Based Weight Assessment
LGBWM	Linear Group Best-Worst Method
LM	Lexicographic Method
LOD	Linked Open Data
MABAC	Multi-Attribute Border Approximation Area Comparison
MACBETH	Measuring Attractiveness by a Categorical Based Evaluation Technique
MADM	Multi-Attribute Decision Making
MAPPAC	Multicriterion Analysis of Preferences by Pair-wise Actions and Criterion Comparisons
MARCOS	Measurement of Alternatives and Ranking according to COmpromise Solution
MARE	Multi-Attribute Range Evaluations
MAS	Multi-Agent Systems
MAUT	Multi-Attribute Utility Theory

Abbreviation	Meaning
MAVT	Multi-Attribute Value Theory
MCDA	Multi-Criteria Decision Analysis
MCDM	Multi-Criteria Decision Making
MDS	Multidimensional Scaling Model
MELCHIOR	Méthode d'ELimination et de CHoix Includent les relations d'ORdre
MEREC	Removal Effects of Criteria
MIN_MAX	MIN_MAX Method
MODM	Multi-Objective Decision Making
Multi-MOORA	Multi-Objective Optimization by Ratio Analysis
N3	Notation 3 RDF format expressed as triples (subject, predicate, object)
NAIADE	Novel Approach to Imprecise Assessment and Decision Environments
NGT	Nominal Group Technique
N-Quads	An extension of N-Triples represented as quadruples (subject, predicate, object, context)
NT	N-Triples
N-Triples	Plain text format for encoding RDF data as a triple of Subject, Predicate and Object
OER	Officer Efficiency Report
OOP	Object-oriented programming
ORESTE	Organization, Rangement Et Synthese De Donnes Relationnelles
OSWMI	Objective-Subjective Weighted Method for Minimizing Inconsistency
OTP	Open Telecom Platform
OWL	Web Ontology Language
PACMAN	Passive and Active Compensability Multicriteria Analysis
PAMSSEM	Procédure d'Agrégation Multicritère de type Surclassement de Synthèse pour Evaluations Mixtes
PRAGMA	Preference Ranking Global Frequencies in Multicriterion Analysis
PROMETHEE	Preference Ranking Organization Method for Enrichment of Evaluations
PSI	Preference Selection Index
QUALIFLEX	QUALIty by FLEXible multicriteria method
R&D	Research and Development
RAFSI	Ranking of Alternatives through Functional Mapping of Criterion Sub-Intervals into a Single Interval
RB	Rule Base
RDF	Resource Description Framework
RDFa	Resource Description Framework in Attributes
RDFLib	RDF Library
RDFS	RDF Schema
Rebar3	Repeatable Erlang Build and Release 3
REGIME	REGIonal Multicriteria Elimination
REMBRANDT	Ratio Estimation in Magnitudes or Deci-Bells to Rate Alternatives which are Non-Dominated

Abbreviation	Meaning
RIF	Rule Interchange Format
RSES	Rough Set Exploration System
RST	Rough Set Theory
SAW	Simple Additive Weighting
SERESYE	Swarm-oriented ERlang Expert System Engine
SIGs	Special Interest Groups
SKOS	Simple Knowledge Organization System
SMART	Simple Multi-Attribute Rating Technique
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SQWRL	Semantic Query-Enhanced Web Rule Language
SSE	Service Selection Advisor
SWARA	Step-Wise Weight Assessment Ratio Analysis
SWRL	Semantic Web Rule Language
TACTIC	Treatment of the Alternatives according to the Importance of Criteria
TODIM	Interactive Multi-Criteria Decision Making
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
TriX	Triple in XML
Turtle	Terse RDF Triple Language
URI	Uniform Resource Identifier
UTA	UTilités Additives
VIKOR	VIseKriterijumska Optimizacija I Kompromisno Resenje
VIMM	Vital-Immaterial Mediocre Method
VM	Virtual Machine
WASPAS	Weighted Aggregated Sum Product Assessment
WME	Working Memory Element
WPM	Weighted Product Model
WSM	Weighted Sum Model
WSM	Weighted Sum Method
WSML	Web Service Modelling Language
WWW	World Wide Web
XML	eXtensible Markup Language
XSD	XML Schema Definition

Chapter 1

Introduction

1.1 Overview

1.1.1 Evolution of Human Resource Management

Human Resource Management (HRM) involves operating several key processes, including the recruitment, selection, training, compensation, and retention of employees, as well as the development of related policies and strategies. Earlier, HRM was mainly focused on administrative tasks such as processing payroll, organizing events, and managing files. Over the past two decades, however, HRM has evolved significantly and has become a strategic role that is vital to the success of every organization. In this context, Human Resources (HR) remain the most crucial capital of any organization, despite the use of modern automated machines and advanced technologies [1]. Moreover, HR can be viewed as an organization's treasure since they are more difficult for competitors to replicate than products and services are. Therefore, recruiting individuals with the right skills for the right jobs can provide significant competitive advantages and substantially contribute to organizational development[2].

1.1.2 Recruitment and Selection in Modern Human Resource Management

Recruitment and selection are crucial processes in HRM that contribute significantly to gaining competitive advantages and organizational growth [3]. The recruitment process focuses on attracting suitable applicants, while the selection process involves choosing the best candidates based on the organization's needs. These processes involve analyzing job requirements, creating job descriptions, developing recruitment methods, attracting applicants, and ultimately evaluating and selecting candidates. Effective recruitment and selection methods result in a better person-organization fit, improved individual and organizational performance, enhanced organization's image, as well as a larger talent pool for organizational growth [2].

In essence, recruitment and selection are dynamic and complex processes facing several challenges, including unreasonable and unrealistic job requirements analysis, discrimination in employment standards, inappropriate recruitment methods, long recruitment and selection cycle, and difficulties in attracting candidates. These challenges can increase costs and complicate the task of finding and selecting the best qualified candidates, thus negatively impact organizational growth [2].

Therefore, it is critical to address these challenges by developing effective recruitment and selection methods that meet specific needs, factors, and positions for different organizations. The effective combination of human expertise and technological solutions is one essential aspect of mitigating these challenges. This mix can result in shorter recruitment and selection cycle, saving time and cost, enhancing candidate pool quality, promoting objectivity, and increasing efficiency, leading to the selection of the best candidates and, in turn, more successful and effective employees and organizations [2].

1.1.3 Erlang's Role in Enhancing Selection Process

This thesis presents a novel approach for enhancing the efficiency of selecting the best candidates for corresponding jobs by developing a job-applicant matching system using the Erlang functional programming language. The approach focuses on integrating the rule engine capabilities of SERESYE[1] (Swarm-oriented ERlang Expert System Engine), the semantic wed data modeling functionality of Semantic Web Toolkit for Erlang Applications[2], and the underlying principles of Multi-Criteria Decision Making (MCDM) using the Weighted Sum Method (WSM), all of which enable comprehensive rule processing and multi-criteria evaluation of applicants qualifications. Since Semantic Web Toolkit for Erlang Applications supports encoding N-Triples (plain text format for encoding RDF data) and JSON (JavaScript Object Notation) data formats of the Resource Description Framework (RDF), the thesis further explores expanding the matching system's data interpretation scope using RDFLib[3]. The RDFLib is a Python library for parsing and processing most of the RDF data formats.

1.1.4 Why Erlang

Trends show that the functional programming paradigm provides advanced abstraction techniques and features to handle complex data structures and concurrency efficiently [4] [5] [6] [7] [8]. It builds modular programs based on the mathematical evaluation of functions and puts emphasis on functions composition, decomposition, and recursion over complex data structures. Such functions often have no hidden side effects, ensuring consistent output given the same input. Other important features of functional programming include immutable data, avoiding shared state, and simplified concurrency [9] [10] [11] [7]. Functional programming languages such as Erlang, Haskell, and F#

[1] https://github.com/afiniate/seresye
[2] https://github.com/fogfish/semantic
[3] https://rdflib.readthedocs.io/en/stable/

intrinsically support concurrency, and have gained increased use and interest in both academic and industrial contexts [6] [12].

Erlang is an open-source and cross-platform functional programming language primarily designed to develop scalable, concurrent, distributed, and reliable real-time systems. Erlang systems are based on the concept of spawning a multitude of concurrent processes that interact asynchronously via message passing and share no memory or state. Such state isolation enhances processes' data safety without introducing significant memory overhead. Erlang also provides dynamic code updating abilities without stopping or interrupting the running system, thereby maintaining high availability. Moreover, Erlang supports features such as immutable data, function composition, decomposition, recursion, and higher-order functions, as well as sophisticated pattern matching over complex data structures. Furthermore, Erlang's distribution is packaged with Mnesia, a distributed database management system, and the generic Open Telecom Platform (OTP) libraries. OTP libraries provide proven solutions known as behaviors that ensure best practices for the rapid development of robust systems. Likewise, Erlang's programming style is influenced by Prolog's logic programming. This makes Erlang a powerful tool for developing advanced expert and control systems [12] [13] [14].

These Erlang features make it a promising technology that can significantly enhance the process of personnel selection. The application of Erlang in this area is yet largely unexplored. This thesis thus aims to exploit such Erlang characteristics and libraries for efficient processing of complex rules and datasets, enabling the development of a sophisticated job-applicant matching and evaluation application.

Furthermore, the feasibility of Erlang is industry proven. For instance, Erlang is used to develop the popular WhatsApp instant messaging system [15]. Moreover, leading organizations such as Facebook, Amazon, Yahoo!, T-Mobile, and Motorola are using Erlang in their systems [14]. The Elixir[4] functional programming language is based on Erlang, and it uses Erlang Virtual Machine (VM) to build scalable systems as well as web and embedded applications. Erlang is also used to develop a number of web frameworks[5], including the Zotonic web framework[6], which uses a fast Erlang-based web server called Cowboy[7]. Several other Erlang applications exist, including RabbitMQ, an advanced message queuing protocol standard; Wing, a 3D graphics modeling

[4] https://elixir-lang.org/
[5] https://github.com/ChicagoBoss/ChicagoBoss/wiki/Comparison-of-Erlang-Web-Frameworks
[6] https://zotonic.com/
[7] https://github.com/ninenines/cowboy

framework; and databases such as CouchDB, SimpleDB, and Scalaris [12], as well as Riak8, a distributed NoSQL database.

In the area of intelligent Multi-Agent Systems (MAS) , Erlang is used to develop a number of MAS including erlang eXperimental Agent Tool (eXAT[9]) [16] [17] [18] [19] [20], Erlang Jason (eJason[10]) [14] [21], Evolutionary-based MAS (EMAS[11]) [22] [23] [24],and others [25]. Intelligent agents concepts emerged from the Artificial Intelligence (AI) field and later form its subfield, Distributed Artificial Intelligence (DAI) [26] [27]. An intelligent agent is a computer system with reactive (event-driven), proactive (goal-directed), and social (communicative) properties that enable it to act autonomously in its environment, ultimately making decisions to achieve its design goals [28] [29] [30] [31] [32] [33].

eXAT is the first Erlang-based agent platform, demonstrating the feasibility of using Erlang to implement all-in-one aspects of scalable and parallel intelligent systems [16]. Moreover, eXAT is used in other applications such as distributed monitoring system [34] and agent migration (eXAT+[12]) [35]. eXAT platform consists of three main parts. First, the agent reactive behavior is programmed using Erlang Finite State Machine (FSM) behavior. Second, agent communication is implemented using Erlang concurrent message passing capabilities complying with the specifications of Agent Communication Language (ACL) which is part of the Foundation for Intelligent Physical Agents (FIPA) standard implemented. Third, agent intelligence is programmed with ERESYE[13] (ERlang Expert SYstem Engine) [36]. ERESYE provides means of building concurrent rule engines, each one with its own knowledge base exploiting, among others, Erlang concurrency, pattern matching, tuples, and function clauses features.

1.1.5 SERESYE: Erlang Expert System

This thesis uses an enhanced version of ERESYE called SERESYE[14] (Swarm-oriented ERlang Expert System Engine). SERESYE maintains all ERESYE core features, but it enhances its performance and scalability in terms of rules organization and propagation. The terms SERESYE and ERESYE are used interchangeably throughout this document, and all research and references relating to ERESYE are equally applicable to SERESYE.

---

[8] https://riak.com/
[9] https://github.com/gleber/exat
[10] https://github.com/avalor/eJason
[11] https://github.com/ParaPhraseAGH/erlang-emas
[12] https://github.com/michalwski/exat
[13] https://sourceforge.net/projects/eresye/ ; https://github.com/TypedLambda/eresye
[14] https://github.com/afiniate/seresye

Furthermore, to align SERESYE with recent versions of Erlang and its associated tools, such as Rebar3, the thesis author undertook several enhancements on Gleber SERESYE[15] version. These enhancements included adjustment to configuration files, incorporation of 'spec' and 'doc' annotations in various modules for better documentation and type specifications, and updates on its supervisor module, as well as server module. As a result, this thesis utilizes the version of SERESY[16] that is updated and maintained by the thesis author, ensuring compatibility with the latest versions of Erlang, and its build tool Rebar3 (Repeatable Erlang Build and Release 3) as well as other dependent libraries.

1.1.6 Semantic Web Toolkit for Erlang Applications

One of the objectives of this thesis is to process semantic web data structures to construct the KB of SERESYE rule engine. Because SERESYE does not incorporate native support for processing semantic web data, this thesis uses Semantic Web Toolkit for Erlang Applications to integrate the processing of semantic web data in the job-applicant matching and evaluation application.

Semantic Web Toolkit for Erlang Applications is a library that defines data models for semantic web data and interprets them in Erlang terms as ground facts or knowledge statements in triple format. It specifically implements semantic codecs for N-Triple, JSON-LD (JavaScript Object Notation for Linked Data), and pure JSON formats. In the context of this thesis, Semantic Web Toolkit for Erlang Applications is used to convert a semantic web RDF ontology representing Business Technology Management (BTM) jobs and applicants specification into Erlang maps data structure. The generated maps are then processed to construct a KB representing the initial asserted jobs and applicants facts in the format of tuples accepted by SERESYE. Based on this KB, a matching rule engine is developed which pattern matches each job fact with corresponding applicant facts and evaluates the applicants qualification based on WSM. Whenever a match is found, a matching rule is fired and its WSM evaluation actions are executed, all of which form new inferred facts that are constructed in match tuples. The rules are designed to evaluate several criteria and the resulting math tuples are added to the KB representing inferred match facts.

Integrating SERESYE with the Semantic Web Toolkit for Erlang Applications offers several key advantages. First, it enhances knowledge representation by utilizing Semantic Web Toolkit for Erlang Applications to interpret data, thereby improving the KB's accuracy and completeness.

---

[15] https://github.com/gleber/seresye
[16] https://github.com/MiloudEloumri/seresye ;
https://github.com/MiloudEloumri/match/tree/main/apps/seresye

Second, this integration enhances rule processing capabilities in SERESYE by offering a more comprehensive KB representation. Third, the challenge of knowledge engineering and representation is also mitigated through the use of semantic web data, addressing a common difficulty in expert systems. Finally, Erlang capabilities for fast, concurrent, and scalable processing enables developing a rule-based matching and evaluation application that is both rapid and scalable.

1.1.7 RDFLib

The research introduces the use of RDFLib[17], a Python library for working with RDF data to facilitate the conversion of RDF formats. The use of RDFLib can increase the scope of data that the system can process and interpret since Semantic Web Toolkit for Erlang Applications supports only N-Triple and JSON formats. RDFLib supports converting several formats including RDF/XML, N3, N-Triples, N-Quads, Turtle, TriX, JSON-LD, RDFa and Microdata. Accordingly, an RDFLib script[18] is used to convert from RDF/XML to N-Triples (nt) format.

1.1.8 Semantic Web

Utilizing semantic web standards and technologies, such as ontologies, can enhance the job-applicant matching process. It provides a standardized way to represent knowledge, making it easier to share and use across different platforms and applications. In this context, the semantic web extends the current web by giving online information a formal meaning, allowing machines to understand and process data more effectively, especially in terms of automation and information exchange. Its foundation relies on four key concepts: expressing meaning, knowledge representation and reasoning, using ontologies, as well as employing intelligent agents for data processing [37]. This has led to the development of several technologies and standards, including the RDF: a standard data model to express knowledge; RDF Schema (RDFS): a simple ontology language that allows describing concepts as metadata models of RDF models; Protocol and RDF Query Language (SPARQL): a standard RDF query language used to query RDF online data stores based on http requests; and Web Ontology Language (OWL): for building ontologies with more semantic and logical expressiveness [38] [39]. Ontologies play a crucial role in the semantic web by providing explicit, formal, and shared domain knowledge models, enabling software systems to interact and share information [38]. This technology combines aspects of symbolic knowledge representation in AI, formal logic, and software engineering, enhancing web functionality.

---

[17] https://rdflib.readthedocs.io/en/stable/
[18] https://github.com/MiloudEloumri/match/blob/main/apps/job_match/priv/data/2-reflib-script.py

Ontology technology has shown increasing research and development in areas such as knowledge representation, methodologies, tools, linked data, semantic search, and intelligent systems, opening new opportunities for advanced application development [33] [38] [40] [41].

1.1.9 Multi-Criteria Decision-Making

To evaluate and score candidates, Multi-Criteria Decision-Making (MCDM) concepts are integrated into the matching system logic. MCDM approaches are used to evaluate and select the most suitable alternatives in the presence of multiple, usually conflicting decision criteria, enabling decision-makers to consider multiple criteria [42]. Hence, given a set of alternatives (options) and various decision criteria (conditions or variables), MCDM aims to offer a selection, ranking, description, classification, or sorting [43]. MCDM encompasses several methods [44]. Among these methods, this thesis uses Weighted Sum Method (WSM), also known as, Simple Additive Weighting (SAW) for its simplicity, relevance, and effectiveness in addressing the complexities of the decision-making scenarios [3] [45] [46].

1.1.10 Practical Applications and Future Scope

Based on the aforementioned technologies, this thesis implemented a prototype named "semantic relatives[19]" for matching and driving family relationships stored in OWL ontology and converted into N-Triple format using RDFLib for further processing by Semantic Web Toolkit for Erlang Applications and SERESYE. The purpose of implementing a simple use case in the domain of relatives is to test the proposed integration approach and gain insights on these technologies. Then, a more complex case study represented by Business Technology Management (BTM) jobs specification is used to match jobs and applicants data along with corresponding evaluation.

1.1.10.1 Business Technology Management Jobs

Designing and developing a rule engine requires identifying and constructing a KB that stores a set of facts representing a domain of interest and rules to derive new facts. As the domain of interest, this thesis considers Business Technology Management (BTM) jobs specification as the data source for the job-applicant matching system.

In 2009, Information Technology Association of Canada (ITAC) introduced BTM initiative in response to business demand for graduates with the right mix of business and technology skills [47]. BTM initiative aims to unify and standardize business and technology educational and

[19] https://github.com/MiloudEloumri/match/tree/main/apps/semantic_relatives

professional aspects. Accordingly, BTM development involves defining several knowledge areas, learning outcomes, competencies, occupations, career paths and accreditation programs based on several related standards. As an educational program, BTM has been offered in several Canadian universities resulting in thousands of graduates [47] [48].

Based on the BTM standardized occupations and using Protégé ontology editor, Ghebli[20] developed BTM jobs ontology incorporating among others actual BTM jobs specifications and examples of imported applicants with randomly assigned qualifications. This thesis applies the matching approach to a modified version of Ghebli's BTM jobs ontology. The modified BTM jobs ontology[21] removes irrelevant concepts that are used in matching in the context of Protégé ontology editor as well as concepts that are not considered in the matching model to simplify the process of constructing the KB and the Rule Base (RB).

1.1.10.2 Future Work

The findings of this thesis illustrate the feasibility of integrating scalable Erlang rule processing with semantic web data structures. Furthermore, the research demonstrates improved job-applicant matching and evaluation results, which can boost organizations' performance and contribute to the areas of recruitment, selection, expert systems, semantic web, and decision-making systems. To enhance the developed matching system, planned future work includes the following: the development of a web interface using Erlang Zotonic web framework, the use of SERESYE multiple parallel rule engines and Erlang Poolboy library[22] for creating and managing a pool of concurrent process and the development of extensions such as emailing, scheduling interviews, and making job offers models.

The following sections discuss the research motivation, significance, problem, objectives, questions, and contribution.

1.2 Motivation and Significance

The motivation of this research is triggered, on one hand, by the crucial need to improve job-applicant matching and evaluation efficiency, a fundamental task in HRM recruitment and selection processes. On the other hand, the research is inspired by the promising features of Erlang

[20] https://github.com/JamalElgebli/BTM-jobs-ontology-prototype
[21] https://github.com/MiloudEloumri/match/blob/main/apps/job_match/priv/data/1-btm-jobs-applicants-ontology.rdf
[22] https://github.com/devinus/poolboy

programming language and semantic web technologies that can be bridged to address this critical need. In this context, the aim is to enhance existing systems by integrating advanced technologies and thus making significant contributions to related business and academic aspects. Following is a more detailed discussion on the motivation and significance of this thesis.

1.2.1 Motivation

Addressing a Critical Need. One of the key practices in HRM is recruiting the applicant whose qualifications best matches the corresponding job requirements. The efficiency and accuracy of a job-applicant matching and evaluation process can significantly impact both the performance and productivity of an organization, as well as the satisfaction, motivation, and the career path of an employee. Therefore, it is important to develop more efficient matching techniques for personnel selection problem.

Improving Existing Systems. Personnel selection methods need to consider multiple criteria simultaneously and adapt new structured data models that can be processed concurrently. This research is, therefore, inspired by the concept of introducing new, flexible, and more efficient perspectives in the realm of a job-applicant matching and evaluation.

Integrating Advanced Technologies. The integration of Erlang technologies represented by SERESYE and Semantic Web Toolkit for Erlang Applications with semantic web standard data in the area of job-applicant matching and evaluation is relatively a new promising endeavor. This thesis is inspired by undertaking such new integration and further investigating its potential applications in the area.

1.2.2 Significance

Business Impact. Matching the right applicant for the right job position results in better satisfaction and improved performance for both the employee and the business. This mutual benefit can collectively contribute to increased productivity leading thus to direct business growth.

Technological Significance. The demand for efficient and modern job-applicant matching and evaluation systems continues to increase as digital technologies trends continue to impact and transform all aspects of businesses including personnel selection. This research aligns with such digital trends by providing significant technological advancement to the challenges of modern personnel selection.

Academic Contribution. This thesis has the potential to contribute and advance the knowledge of related academic fields including decision-making systems, expert systems, semantic web technologies, and HRM selection process. By introducing and developing a new approach to job-applicant matching dilemma, this research can serve as a foundation for further research and innovation in these academic fields.

Practical Implications. The findings of this research convey significant practical implications. The developed job-applicant matching and evaluation model has the potential to transform how organizations practically perform selection operations. Furthermore, it can serve as a motivation for other researchers to apply related technologies in a variety of contexts, therefore, expanding the scope of research impact and practical applications.

1.3 Research Problem

Despite technological advancements, matching applicants with job requirements efficiently remains a challenging personnel selection problem. Traditional methods often lack the ability to simultaneously process multiple criteria, comprehensively evaluate numerous numbers of applicants, and effectively operate concurrently on well-structured data and rules . The complexity of such matching and evaluation problem calls for more sophisticated approaches. Moreover, existing systems may not fully leverage the capabilities of expert systems such as SERESYE integrated with Semantic Web Toolkit for Erlang Applications to process semantic web standard data structures and evaluate applicants based on MCDM principles . Therefore, there is a need to explore how to effectively integrate these technologies and apply them to the personnel selection problem. The challenge, therefore, is to develop and test a job-applicant matching and evaluation system that uses these technologies to deliver an effective solution to the complex problem of personnel selection. The matching and evaluation system should operate on large datasets and evaluate the qualifications of the applicants against the requirements of the corresponding job based on MCDM methods. It should also provide flexible results querying mechanism based on job(s), applicant(s) or other querying combinations.

1.4 Objectives and Research Questions

Building on the capabilities of Erlang's SERESYE and Semantic Web Toolkit for Erlang Applications, this research aims to advance the process of personnel selection. To accomplish this aim, the following objectives and corresponding research questions are identified:

Objective 1: The initial objective is to critically examine the current state of the art of personnel selection methods to thoroughly understand the key challenges and limitations that affect the efficiency of the existing methods.

Research Question 1: What are the key challenges and limitations in the current personnel selection methods?

Next, the research considers the Business Technology Management (BTM) jobs-applicants ontology as the dataset for the matching and evaluation application.

Objective 2: This objective includes understanding and analyzing the structure and the patterns of BTM jobs-applicants ontology stored in RDF format.

Research Question 2: How can BTM jobs-applicants ontology be effectively analyzed and used as dataset in the job-applicant matching and evaluation application?

Following the critical study on the existing methods and the analysis of BTM jobs-applicants ontology, the research shifts to focus on the prospective solution based on Erlang's SERESYE and Semantic Web Toolkit for Erlang Applications.

Objective 3: This objective involves studying the capabilities of SERESYE and Semantic Web Toolkit for Erlang Applications to identify possible integration and development approaches.

Research Question 3: How can SERESYE and Semantic Web Toolkit for Erlang Applications be applied to enhance the job-applicant matching and evaluation process?

Based on the studies and analyses conducted on the previous objectives, the research carries the design and implementation phase.

Objective 4: The fourth goal includes designing and implementing a job-applicant matching and evaluation model that integrates SERESYE, Semantic Web Toolkit for Erlang Applications, and the principles of MCDM utilizing WSM. The aim is to develop a model capable of evaluating multiple criteria in order to obtain a more accurate matching result.

Research Question 4: How can an effective job-applicant matching and evaluation model be designed and implemented based on the integration of SERESYE, Semantic Web Toolkit for Erlang Applications, and MCDM using WSM?

The research then moves to the evaluation phase.

Objective 5: This fifth objective considers evaluating the results of the devolved model.

Research Question 5: What are the results findings and limitations?

Based on the obtained experience and the results of the developed model, the research finally concludes the findings and explores potential future work.

Objective 6: The final objective involves stating the research findings, discussing the limitations, and identifying potential future enhancements.

Research Question 6: What are the potential areas of improvement and future directions for the developed model?

These objectives and research questions serve as the thesis' roadmap, leading its literature review, methodology, design and implementation, results and testing discussion, conclusion, limitation, and future work.

1.5 Contribution

This thesis makes significant contributions by providing new insights to the challenging problem of personnel selection. It stands out by the integration of advanced technologies and decision-making concepts resulting in a unique matching and evaluation model based on the capabilities of Erlang, semantic web technologies, and MCDM principles. Moreover, this research identifies the limitations of the developed matching model and outlines potential future enhancements to encourage continuous progress addressing this problem. These contributions are detailed as follows:

Technologies Integration. The main contribution of this thesis is the innovative method of integrating various technologies aimed at enhancing the efficacy of the job-applicant matching and evaluation process. In particular, this research proposes a unique concepts synthesis represented by the rule processing over semantic web KB and multi criteria evaluation. This novel integration not only offers a new perspective for improving the efficiency and accuracy of the job-applicant matching and evaluation but also opens the opportunities for conducting further related studies.

Matching Model Development. The actual development of a new model for job-applicant matching and evaluation is a core part of this research contribution. Since the model is based on the foundation of expert systems, semantic web technologies, and decision-making theories, it has the potential to overcome the limitations present in existing methods, thus providing a more effective solution to the personnel selection challenge. This not only adds to the existing body of related literature but also offers interested parties a powerful matching system for improving recruitment operations.

Semantic Web Data Application. This research investigates relatively unexplored concept of utilizing semantic web data standards in the area of personnel selection and processing it using

Erlang technologies. This application enables bridging the gap between complex semantic web data structures and production rule processing, therefore contributing to the scope of improvement in the area.

MCDM Principles. This study introduces the use of WSM as a specific application of MCDM in the context of job-applicant matching and evaluation. This introduction allows evaluating multiple criteria simultaneously. Such use of MCDM adds considerable advancement to the existing techniques, hence expanding the range of feasible solutions in the area.

Testing and Validation. Another essential contribution of this study is the robust testing and validation of the developed matching model using case studies represented by the simple case of family relationship and a more complex case of BTM jobs-applicants matching. This testing and validation allow us to reveal new perspectives and provide useful feedback for potential improvement.

Future Enhancement. Identifying potential improvement and future work for the developed matching model is an important contribution of this research. By discussing the limitations and future directions, the research opens up further innovative exploration in the area and provides a foundation for subsequent research.

1.6 Thesis Structure

Chapter 1 overviews involved topics and provides thesis motivation, significance, research problem, objectives, and contributions.

Chapter 2 delves into the practices and processes of HRM, with a particular focus on the recruitment and selection processes concepts, methods and challenges.

Chapter 3 then introduces Erlang programming language and expert systems, with a focus on Erlang-based expert systems. The chapter also includes literature review on the application of expert systems in the personnel selection problem.

Chapter 4 provides background on semantic web technologies, focusing on ontology concepts and applications, as well as the Semantic Web Toolkit for Erlang Applications.

Chapter 5 provides a comprehensive background on MCDM concepts and methods. The chapter also includes literature review on the application of MCDM methods in personnel selection.

Chapter 6 discusses the research methodology. It starts by overviewing the generic Design Science Research Methodology (DSRM). It then outlines the specific design and development approach used to develop the job-applicant matching and evaluation application. Furthermore, the chapter includes detailed descriptions of the BTM jobs-applicant ontology dataset and its analysis as well as the approach of processing the ontology data using RDFLib and then the Semantic Web Toolkit for Erlang Applications. Then, the chapter explains the selection criteria and evaluation concepts based on WSM, followed by the approach of constructing the SERESYE rule engine KB and the implementation logic of the matching rules.

Chapter 7 provides in-depth discussion on the foundation and practical implementation of the job-applicant matching and evaluation application. This includes details on WSM formulas, selection and evaluation criteria, and the design and implementation of the application modules.

Chapter 8 discusses the results of the application and the research. It covers the running of the job-applicant match application, querying the rule engine results, fine-grained queries and observations and enhancements based on the results. The chapter then discusses the research objectives and the answers to the corresponding research questions.

Chapter 9 concludes the findings, discusses the limitations of the study, and outlines potential enhancement and future work.

Each chapter in this thesis builds upon the previous ones, creating a coherent and comprehensive exploration of the intersection between HRM practices, Erlang expert systems, semantic web technologies, and MCDM methods in the context of job-applicant matching and evaluation addressing the personnel selection challenge.

Chapter 2

Human Resource Management

2.1 Overview

Human Resource Management (HRM) involves employing individuals, training them, and compensating them, as well as developing policies and strategies to manage and retain them in the organization. Earlier, HRM was perceived as an administrative role responsible for performing tasks such as processing payroll, coordinating organization trips, and managing forms and files. During the past two decades, however, HRM's role has evolved significantly and has become a major part of every organization's strategic planning [1] [3].

Human resources (HR), also known as intellectual capital, are considered the most valuable asset of organizations as they provide a competitive advantage that is harder for competitors to replicate compared to goods and services. Employees who demonstrate proficiency in their professional expertise and align with the organizational values and culture significantly contribute to the overall organization's success. Thus, this emphasizes the critical role of HRM in employing the right personnel and, in turn, maintaining competitive advantage and organizational growth. Through a merit-based hiring and selection system, organizations can maximize the efficiency of their workforce. Such systematic selection of personnel not only enhances operational efficiency but also improves the organization's position in the market, hence setting the stage for sustainable productivity and competitive advantages [2] [3] [49].

2.1.1 Human Resource Management Main Tasks

HRM typically involves managing seven top tasks, which are staffing, workplace policies, compensation and benefits, retention, training and development, employment laws, and worker protection. To perform these tasks successfully, it is essential for HRM to maintain excellent communication and management skills, as well as recognize external factors that may positively or negatively impact the organization. Some examples of external factors include globalization, crises (e.g., COVID), employment laws, healthcare costs, and technological changes. Accordingly, HRM needs to understand and analyze the dynamics of these forces, which typically requires reviewing reliable and related publications as well as attending conferences that discuss such changes and issues. This allows human resources managers to make strategic decisions and develop policies addressing both the organization's and employees' needs, thereby sustaining a legal, motivated, and productive workplace. Figure 1 depicts the main tasks of HRM, followed by a brief description of each task [1].

Figure 1 Human Resource Management Main Tasks

2.1.1.1 Staffing

Even with the use of modern automated machines and advanced technologies, HR remain the most valuable capital of any organization [1]. They play a crucial role in accomplishing tasks and achieving work objectives; therefore, staffing is a key task in HRM activities. Staffing encompasses the entire process of hiring, from job postings to salary negotiations. It commonly consists of four main steps, as outlined in the following list:

1. **Staffing plan development**. This step involves creating a staffing plan to identify the needed personnel based on several factors.
2. **Multiculturalism policy development**. Supporting multiculturalism at work has become increasingly important, thus, this step ensures creating policies that value individuals from different backgrounds.

3. **Recruitment**. This step involves finding the right people to fill the available positions using various recruitment strategies.
4. **Selection**. This step includes interviewing, and selecting the best-qualified candidates, as well as negotiating compensation using several methods. It is then followed by training, retention, and motivation plans.

2.1.1.2 Workplace Policies

Workplace policies are important for ensuring equality and consistency within an organization, and HRM plays a key role in the development of these policies. The policy development process involves collaboration between management, executives, and HRM professionals. For instance, HRM professionals typically recognize the need for a policy or a change in a policy and accordingly collect views, write the policy, and then communicate it to employees. Examples of workplace policies include discipline processes, vacation time, dress codes, ethics, and internet usage policies. In such policy development, it is essential to note that the HR department does not work separately but instead collaborates with all other departments in the organization [1]

2.1.1.3 Compensation and Benefits

HRM professionals need to ensure that compensation is fair, meets related standards, attracts employees, and compares to similar jobs in other organizations. To meet this need, it requires setting up compensation systems that account for years of experience, education, and other factors. Compensation examples include salary, health benefits, retirement plans, bonuses, and tuition reimbursement [1].

2.1.1.4 Retention

Retention is the process of encouraging employees to remain in the organization. While compensation is an important factor in keeping employees, other factors that influence employee retention exist. Job-related, culture-related, and workplace-related issues, as well as conflicts with managers, are the reasons causing 90% of employees to leave organizations. Despite these main causes, 90% of managers believe that employees leave because of compensation-related issues. Consequently, managers often change compensation packages in an attempt to retain employees, even though compensation may not be the actual reason causing employees to quit [1].

2.1.1.5 Training and Development

Training and development programs ensure continuous professional and personal growth as well as higher productivity. Equally important, they help with employee satisfaction, motivation, and retention. Examples of training programs include skills, communication, teamwork, policy, and legal training [1].

2.1.1.6 Employment Laws

HRM must realize all the laws affecting the workplace, including discrimination, healthcare, compensation, safety, and labor laws. Since the legal environment of HRM is consistently changing, HRM needs to continuously recognize these changes and communicate them to the management of the entering organization [1].

2.1.1.7 Worker Protection

In all organizations, safety is a main concern that requires the frequent establishment of laws at the federal and state levels ensuring worker protection. Unions and union contracts can also influence worker safety regulations. Human resource managers, therefore, need to understand worker protection requirements and guarantee workplace compliance with all standards. Worker safety concerns may include chemical risks, heating and ventilation requirements, the use of “no fragrance” zones, and privacy protection. In some industries, understanding safety regulations is vital, as they can save lives or lead to death [1].

2.1.2 Human Resource Management Main Challenges

HRM faces several challenges, among which the major ones are cost containment, technology dynamics, economic trends, workforce changes, and ethics policies [1].

2.1.2.1 Cost Containment

Cost containment is one of the biggest modern challenges in HRM, and it can determine the success or failure of businesses. Containing costs is a challenge that requires finding the right balance between the offered benefits and their impact on motivating employees or prospective candidates. In addition to other expenses, costs considering the employee's part include healthcare, training, recruiting, and turnover (turnover refers to the number of employees who leave a company during a specified time period). In particular, recruiting and turnover can be very expensive. For instance, studies show that the cost of recruiting a new employee or replacing a turnover can be as high as $9,777 for a position paying $60,000. Therefore, it is essential to develop an efficient and proper recruiting process that can save costs and ensure employee retention. One factor that can contribute to reducing turnover and increasing employee motivation is a proper recruitment and selection process that ensures hiring the right applicants for the right job the first time [1].

2.1.2.2 Technology Dynamics

HRM practices are constantly impacted by the dynamics of technology. For instance, technology has enabled mobile, diverse, and virtual workforces, as well as the use of smartphones, and social networking. These technological changes require addressing a unique challenge by exploiting the advantages of the technology without negatively impacting productivity but instead increasing it. Moreover, technology has provided HRM with a variety of databases to perform various tasks such as tracking employee data, compensation, and training, as well as tracking recruiting processes [1]

2.1.2.3 Economy Trends

Economic status has a significant impact on HRM practices. For instance, economic decline commonly results in high layoffs, introducing issues on a country and business level as well as challenges and restrictions specific to HRM represented by performance and legal issues. In turn, massive recruiting to meet demands during economic growth may occur. In both cases, it is important to develop effective recruitment approaches to meet such economic trends and ensure the right number of workers are present at the right time, as well as consider related legal implications[1].

2.1.2.4 Workforce Changes

HRM faces the challenge of a constantly changing multigenerational workforce. Therefore, HR managers need to address various generations needs and develop corresponding benefits, compensation, and social security legislation plans. Moreover, the retirement of baby boomers requires the replacement of their knowledge and experience as part of strategic planning that accounts for monitoring current workers' skill levels and retirement periods as well as predicting future workforce needs [1].

2.1.2.5 Ethics Polices

HRM departments are responsible for designing codes of ethics and developing policies for ethical decision-making. Therefore, HR managers need to understand the various ethical challenges faced by employees and promote ethics and integrity in the workplace. A code of ethics outlines the expected ethical behavior of employees and may include penalties for ethics violations. Many organizations hire ethics officers specifically to address these important challenges [1].

2.2 Thesis Scope in the Human Resource Management Context

HRM is a vast area encompassing several research topics and practices. This thesis is not particularly holding HRM as a research area in a whole, rather it touches on and investigates a small part of it related to recruitment and selection processes in a more applied approach as opposed to its management, administrative, and legal aspects, focusing on the foundation of technological solutions to the challenge of selecting the right people for open positions, in other words, matching applicants qualifications with job requirements.

Given the previous brief HRM overview, this research moves forward in discussing the background and literature on the job-applicants matching or selecting challenges subsequently throughout the thesis. This research main focus is primarily on: (1) literature discussing job requirements and applicants qualification to analyze and extract main components; identify patterns or standards; and formulate evaluation criteria to build a proper procedure incorporating the identified matching technique under the technological approach of the research; and (2) related methods implementing job-applicants matching and evaluation models, especially those using expert systems and MCDM methods.

2.3 Recruitment Process

The recruitment process provides a pool of applications from which organizations during the selection process, select the best applicants matching the job requirements [1]. Recruitment is an important and challenging HRM task requiring thorough strategic planning. Before recruiting, organizations must develop staffing plans and forecasts to determine their needs. Forecasting is based on several internal and external factors. Internal factors involve budget constraints, short-term and long-term plans, the organizational life cycle, expected turnover, production levels, sales trends, and global expansion plans. External factors include technological change, changes in laws, unemployment rates, population shifts, urbanization, suburbanization, rural changes, and competition. After collecting and analyzing forecasting data, HR professionals can identify recruitment gaps, analyze job requirements, create job descriptions, develop recruitment strategies, and then initiate the recruitment process to ultimately receive a pool of applications ready for the selection or interviewing process [1]. The subsequent sections provide more details on the main tasks and concepts involved in the recruitment process.

2.3.1 Finding the Right Talent

Recruiting the right talent for the right position at the right time requires skill, practice, and strategic planning. To achieve strategic and successful recruitment, it is essential to understand the job market and the factors impacting it. In particular, a typical recruitment process involves completing the following tasks [1]:

1. Consulting the staffing plan.
2. Verifying the accuracy of the job analysis through questionnaires or other means.
3. Creating the job description and specifications.
4. Using a bidding system to recruit and evaluate internal candidates for potential promotions.
5. Determining the most effective recruitment strategies for the specific position.
6. Implementing recruitment strategies.

The recruitment process starts with acknowledging a job opening by referring to the staffing plan, then conducting a job analysis and creating a job description and specification. Next, Using an internal bidding system or a job posting procedure, internal candidates are first encouraged to apply for the position, and if a candidate is accepted, the job may not be externally published. However, publishing open positions within and outside the organization ensures diversity. Then, the best recruitment strategies for the position are determined. For instance, hiring an external head-hunting firm for a high-level executive position could be the preferred recruitment strategy, while advertising on social media platforms might be the best strategy for an entry-level position. Moreover, factors such as legal considerations, deadlines, a low number of applications, and saving time by establishing a system for processing applications and resumes should be considered when developing and managing the recruitment process. Organizations commonly use a mix of strategies to manage the recruitment process and reach the optimal recruitment outcome. The aim of the recruitment process is to find a diverse pool of applicants ready for the interview phase, also known as the selection process. Figure 2 depicts the main tasks in the recruitment process [1].

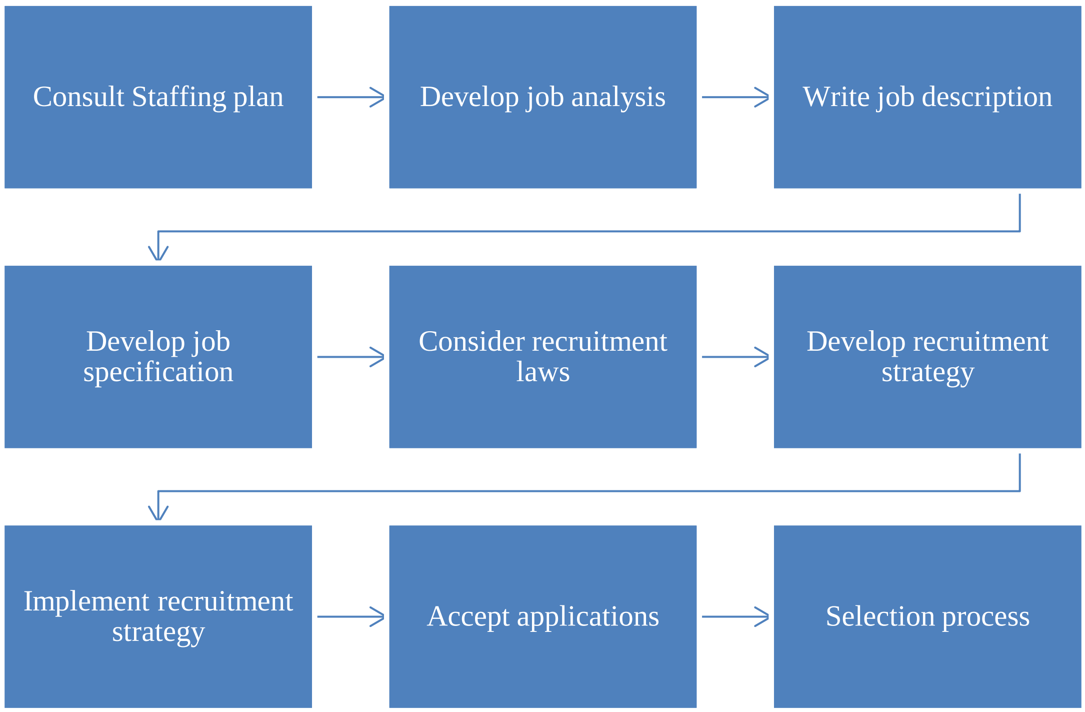

Figure 2 Recruitment Process Main Tasks

2.3.2 Understanding Job Analysis and Description

A job analysis is a formal system used to determine the specific tasks that are required to successfully perform a certain job. Its main purpose is to ensure a right fit between the job and the employee, as well as to establish criteria for evaluating employee performance. To achieve effective job analyses, it involves conducting various types of research, including reviewing current employees responsibilities, studying competitors similar job descriptions, and analyzing any new responsibilities associated with the position [1].

In the context of job redesign, Hackman and Oldham recommended the use of a job diagnostic survey to analyze job characteristics before any job redesign [50]. The authors introduced a model of job design that combines job and individual characteristics to predict when job redesign positively affects employees. They examined the complex interaction between job design and motivation. Understanding job characteristics that influence motivation can help organizations create job descriptions that attract suitable applicants as well as select motivated applicants. It should be noted that job analysis is different from job redesign. Job redesign refers to modifying job requirements for improved efficiency, such as changing job tasks based on new technological advancements [1]. Figure 3 summarizes the process of writing job analyses [1].

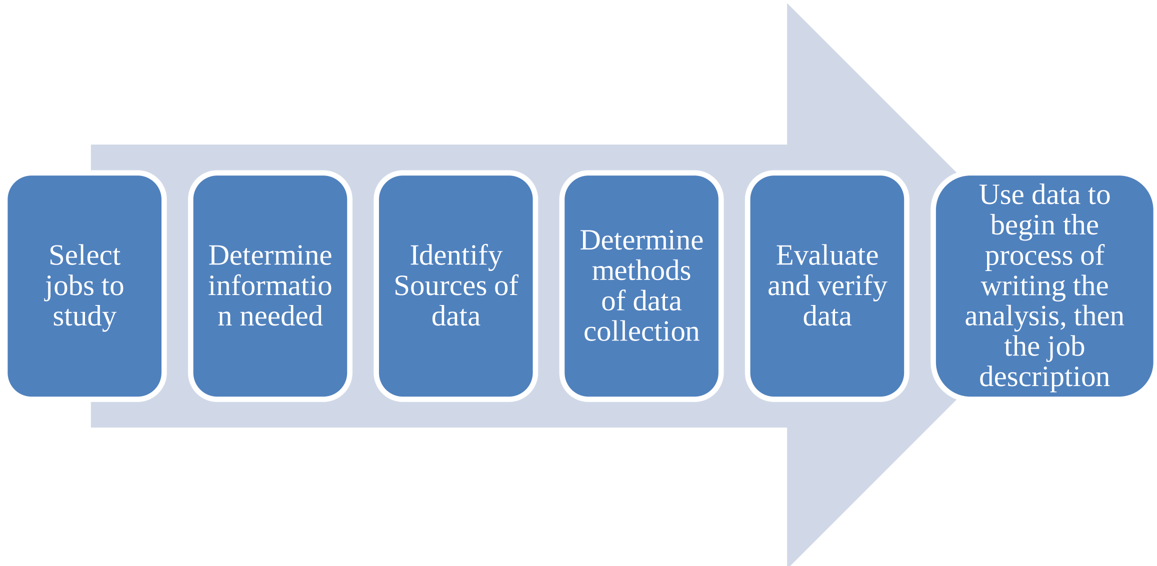

Figure 3 Process for Writing Job Analysis

The obtained data are used to write the job analysis, based on which the job description and specification are created. A job description is a list of tasks and responsibilities associated with a particular job, while job specifications describe the skills and abilities required to perform the job. Job descriptions and specifications are intrinsically linked, consequently job descriptions are frequently designed to include job specifications. The focus on tasks or skills determines the type of job analysis: task-based or competency-based analysis [1].

2.3.2.1 Task-Based and Competency-Based Job Analysis

Job analyses can be conducted using two approaches: task-based analysis or competency-based analysis. A task-based analysis focuses on the tasks and duties of the job, while a competency-based analysis focuses on the knowledge and skills an employee must demonstrate to perform the job. A task-based analysis lists required tasks and duties such as writing performance evaluations, preparing reports, answering phone calls, and assisting customers, while a competency-based analysis includes skills such as being able to use data analysis tools, being able to work within teams, being adaptable, and being innovative. These two types of analyses are used for different purposes and are suited for different job types. While task-based analyses are more objective and clearly list specific tasks, competency-based analyses are more subjective as they describe how a person can apply their skills to perform the job. Competency-based analyses are more suitable for higher-level positions. For instance, a task-based analysis can be used for a receptionist position, while a competency-based analysis can be used for a vice president of sales position. Since competency-based analyses are more subjective, it is more challenging to evaluate whether a candidate meets the required criteria or not. Moreover, legal implications should be considered when choosing job analysis types [1]

2.3.2.2 Conducting Job Analysis

When preparing to write job analyses, it is important to involve managers and make the analyses useful at all levels of the organization. This allows analysts to decide whether to conduct analyses for all positions or focus on specific departments. Furthermore, a tool to conduct the analysis is often used. Questionnaires, for example, either online or in hard copy, are frequently used to determine the duties of each job. Moreover, face-to-face interviews with existing employees are occasionally used instead of questionnaires, but conducting such interviews largely depends on time constraints and the size of the organization. In the context of questionnaires, A typical job analysis questionnaire includes questions about employee information (e.g., job title, how long in position, level of education, and years of experience), key tasks and duties, decision making and problem-solving questions, contacts (e.g., with colleagues, managers, outside vendors, and customers), physical demands, personal abilities, required job skills, and certifications. After employees complete the questionnaires, the collected data can be organized to assist in the creation of job descriptions. If multiple employees complete the questionnaire for the same job, their data is combined to create a comprehensive job analysis. Likewise, software packages such as AutoGOJA can assist HRM departments in performing this task [1]

2.3.2.3 Creating Job Description

The next step after completing the job analysis is writing the job description based on the collected data. A job description typically includes tasks and requirements such as knowledge, skills, abilities, education, and experience, as well as physical requirements. Once the job description is created, HR professionals must obtain approval from the hiring manager before initiating the recruitment process. Moreover, it is important to consider related laws and their implications in the recruitment process [1].

2.3.3 Recruitment Legal Considerations

The law plays a crucial role in all HRM activities, and the recruitment process is not an exception. HR professionals must adhere to legal requirements and ensure a fair and inclusive recruitment process. Based on a country's laws, it is also essential to verify the eligibility of applicants to work in the country. Therefore, the recruitment process often includes in its application questions regarding work eligibility, and HR professionals are responsible for verifying the eligibility of all applicants. Moreover, HR professionals must ensure equality in the recruitment process by adhering to anti-discrimination laws. These laws prohibit discrimination based on gender, race, religion, disability, and other factors. Job announcements typically include equality statements, and HRM is obliged to display notices of equality in visible areas of the workplace [1].

2.3.4 Recruitment Methods

Following the development of the job analysis and job description, A strategic plan is crucial to implementing a successful recruitment process. This plan typically specifies methods of recruitment and expected timelines. Moreover, the plan usually considers diverse recruitment methods to ensure a diverse pool of applicants. It also considers economic situations such as receiving hundreds of applications during high unemployment periods or not receiving enough applications in good economic times. Recruitment methods include hiring recruiters who network and attend events, utilizing campus recruiting programs, using professional associations, utilizing websites and social media platforms, participating in events and job fairs, targeting specific interest groups, and implementing referral programs. These methods are briefly described in the following sections [1].

2.3.4.1 Recruiters

Recruiters are individuals or firms that focus on providing recruitment services. Recruiters are skilled at networking and often attend events to attract potential candidates. They also maintain a constant record of potential candidates in case a suitable future position becomes available. There are three main types of recruiters [1]:

1. **Executive search firms:** These firms focus on high-level positions, such as management and CEO roles. They typically charge a fee of 10–20% of the first year's salary, but they handle much of the upfront work and send qualified candidates who meet the qualifications.
2. **Temporary recruitment or staffing firms:** These firms focus on temporary positions. For example, if an employee is going on medical leave, a temporary recruitment firm can provide a qualified replacement willing to work on a short-term contract. The firm pays the salary of the employee, and the company pays the recruitment firm. If the temporary employee performs well, there may be opportunities to offer them a permanent position.

3. **Corporate recruiter:** A corporate recruiter is an employee within a company who focuses only on recruiting for that company. They may specialize in a specific area, such as technical recruiting.

Recruiters play a crucial role in the recruitment process, but HR professionals are still responsible for managing the process and the recruiters. This includes developing a job analysis and description as well as conducting candidate interviews [1].

2.3.4.2 Campus Recruiting

Campus recruiting involves attracting candidates from colleges and universities, especially for entry-level positions. Campus recruiting programs are often used to develop new talent that has the potential to grow with the organization. Moreover, organizations often establish relationships with campus communities, such as career services departments, and attend campus events such as job fairs. For instance, IBM has successful campus recruiting programs that ensure a large pool of candidates to support its growth. Furthermore, some organization offer student internship program that can lead to full-time employment and cost-saving opportunities [1].

2.3.4.3 Professional Associations

Professional associations are nonprofit organizations that aim to advance a specific profession. Most professions have their own professional organization. For example, in the field of human resources, the Society for Human Resource Management allows companies to post HR-related job openings. Such associations may require paid membership to post jobs. Furthermore, labor unions can also be a good recruitment strategy, and some unions allow job postings on their websites. To effectively utilize professional associations as a recruitment strategy, it is important to identify related organizations and establish relationships with them. This networking helps in connecting with qualified individuals for potential job opportunities [1].

2.3.4.4 Online Job Platforms

In addition to the company’s own website, there are several popular job websites, such as Indeed and LinkedIn, most of which offer inexpensive job postings. However, one of their disadvantages is the large number of resumes that may be received, some of which may not be qualified. To address this issue, many organizations use software that searches for keywords in resumes to identify qualified candidates [1].

2.3.4.5 Social Media

Social media platforms can be used to attract a variety of candidates. Sodexo, a company that provides food service and facilities management, started using social media in 2007 to spread its culture and saved $300,000 on traditional recruiting methods. Similarly, Zappos uses YouTube videos to promote their jobs and culture. Likewise, Facebook Marketplace allows for free job postings, and additionally the business's Facebook page can also be used as a recruiting tool. While Twitter may be a good approach to recruiting people who are open about their job search, LinkedIn can be a better way to find more experienced candidates who are unable to be open about their job search due to their current employment situation because LinkedIn allows users to post their resumes without fearing to lose their current job. A social media recruitment strategy is relatively inexpensive and it is used to promote an organization, share success stories, and highlight culture. However, it is critical to demonstrate the business's culture early by interacting with users before they consider the business an employer [1].

2.3.4.6 Events

Many organizations, such as Microsoft, hold annual events that provide opportunities for networking and learning about new technologies. For example, Microsoft's Professional Developer Conference (PDC) attracts thousands of web developers and professionals seeking to update their skills and expand their professional network. Participating in job fairs is also an effective way to meet a large variety of potential candidates. Additionally, attending generic industry-related events provides opportunities to meet qualified candidates [1].

2.3.4.7 Special Interest Groups

Special or Specific Interest Groups (SIGs) are groups that may require membership and concentrate on certain topics for their members. Jobs can be posted in specific areas on SIGs or listed on discussion boards. For instance, the Women in Project Management SIG offers a section for job postings. Recruiting using SIGs is an effective way to target a specific group of candidates who are trained in a particular field or have a specific specialty [1].

2.3.4.8 Referrals

The referral strategy is effective and commonly results in quickly hiring highly qualified individuals. Organizations usually send job openings to current employees and offer them rewards for successful referrals. Because formal referral programs are generally successful, it is suggested to incorporate them into the overall HRM strategic plan. However, using only referrals can result in a lack of diversity and an increase in nepotism[1].

2.3.4.9 Advantages and Disadvantages of Recruiting Methods

Table 1 highlights the advantages and disadvantages of the discussed recruitment methods [1].

Table 1 Advantages and Disadvantages of Recruiting Methods

Recruitment Method	Advantages	Disadvantages
Outside recruiters	Time saving	Expensive Less control over final candidates to be interviewed
Campus recruiting	Employment growth with the organization Source of talent	Time consuming Suitable for certain types of experience levels
Professional associations	Industry specific Networking	Ads fee Time consuming to network
Online Jobs Platforms	Diversity friendly Low cost Quick	Too broad Hundreds of resumes
Social media	Inexpensive	Time consuming Overwhelming response
Events	Access to specific target markets of candidates	Expensive Right target market consideration
SIG	Industry specific	Research required for specific SIGS tied to jobs
Referrals	Higher quality people Retention	Diversity concerns Nepotism
Unsolicited resumes and applications	Inexpensive, especially with time-saving keyword resume search software	Time consuming
Internet and/or traditional advertisements	Target a specific audience	Expensive
Employee leasing	Less compensation and benefits administration Alternative to temporary employment	Possible costs Less control of who interviews for the position
Public employment agencies	Diverse workforce Less or no costs Service points	Hundreds of resumes Time consuming

Recruitment Method	Advantages	Disadvantages
Labor unions	Access to specialized skills	Not applicable to some jobs or industries Builds relationship with the union

2.3.5 Recruitment Cost

In Recruitment planning, it is essential to allocate a budget for the costs associated with acquiring a pool of applicants. This includes advertising, recruiters, referrals, social media, and event costs. It is also important to calculate the yield ratio. The yield ratio is the percentage of applicants from a particular recruitment method who progress to the selection process for interviewing. This helps in identifying the best recruitment method for specific positions. Yield ratios, however, vary depending on the job and may come from expensive methods [1].

2.4 Selection Process

After implementing the recruitment plan and attracting a pool of potential candidates, the selection process starts. The selection process refers to the steps involved in selecting a candidate with the right qualifications to fill a current or prospective job opening. Typically, managers and supervisors are those who are ultimately responsible for hiring individuals, but the role of HRM is to define and perform several tasks including reviewing resumes, evaluating applications, testing candidates, and conducting interviews. These tasks are expensive and time-consuming. According to the US Department of Labor and Statistics estimation, the total direct and indirect cost of hiring a new employee can reach up to $40,000. Moreover, the Austin, Texas, fire department estimated that it would cost $150,000 to reinterview candidates due to the disclosure of interview questions to the public, which may have introduced unfair advantages to certain candidates. Consequently, it is important to recruit the right candidate from the start and ensure a fair selection process [1].

2.4.1 Selection Process Steps

The selection process consists of five main steps, briefly explained in the following list, and highlighted in Table 2 [1].

1. **Criteria development.** Criteria development involves determining sources of information and ways of evaluating them during the interview. These criteria are typically directly related to the job analysis and descriptions, such as experience, skills, and abilities, as well as aspects of personality or cultural fit. For example, criteria for a project management job might include the following:

- Two years of experience managing a $2 million or more project budget,
- A bachelor's degree in business or a closely related field,
- Ability to work on multiple projects at once,
- Problem-solving ability,
- Conflict-management ability,
- Ability to manage a team of five to six diverse workers,
- Score of at least 70 on the cognitive ability test,
- Score of excellent from the most recent employer.

By developing criteria, the required qualifications become clearer, which makes it easier to determine candidates who move forward in the selection process. For instance, if a bachelor's degree is a criterion and a candidate does not have the degree, their application can be excluded or considered for another job opening.

2. **Application and resume review.** Applications and resumes are reviewed using various methods, such as software tools (e.g., Sovren software) that search for keywords to filter and narrow down the number of resumes to be considered in interviews, thus saving time. Such software tools should be useful and reliable to measure candidate attributes accurately and consistently for a specific job opening. A tool may include features such as resume scanning, cognitive tests, work samples, credit reports, Biographical Information Blanks (BIBs), weighted application forms, personality tests, and interview questions.

3. **Interviewing.** After filtering and identifying the applications that meet the minimum criteria, candidates are selected for interviewing. However, further filtering due to time constraints is usually done through phone interviews.

4. **Test administration.** Before making a hiring decision, various tests are administered, such as drug tests, physical tests, personality tests, and cognitive tests. Moreover, some organizations conduct reference checks, credit report checks, and background checks. These tests are conducted after narrowing down the pool of candidates.

5. **Making the offer.** The last step in the selection process is to offer the job position to the selected candidate. It involves making a formal offer via email or letter defining the compensation and benefits associated with the position.

Table 2 Selection Process Steps Overview

Step	Tasks
Criteria Development	• Understand Knowledge, Skills, Abilities, and Other characteristics (KSAOs) • Determine sources of KSAOs information such as testing, interviews • Develop scoring system for each of the sources of information. • Create an interview plan
Application and Resume Review	• Consult the developed criteria in step one • Consider internal versus external candidates
Interview	• Determine types of interviews • Write interview questions • Avoid interview bias
Test Administration	• Perform testing based on criteria development • May include reviewing work samples, drug testing or cognitive tests
Selection	• Determine which selection method to use • Compare selection method criteria
Making the Offer	• Use negotiation techniques • Write the offer letter or employment agreement

2.4.2 Selection Methods

The selection methods used in the selection process vary in terms of techniques and objectives. Traditional selection process methods can be categorized in two top categories: clinical and statistical techniques [1].

2.4.2.1 Clinical Selection Method

In the clinical personnel selection method, decisions are mainly based on the reviews of the decision-makers. The clinical selection method has roots in psychological science and uses techniques such as work sampling, interviews and various tests including ability, personality, and job knowledge tests. Relying only on this method can lead to biases as it is subject to personal opinions, which may result in errors and different treatment based on factors such as age, race, or gender. To address these issues, scientific approaches are incorporated in the personnel selection process. These approaches include the use of MCDM methods, expert systems, grey relational models, and hybrid models [1] [3] [49].

2.4.2.2 Statistical Selection Method

In the statistical method, a selection model is developed that assigns weights and scores to different factors based on their importance for the job. This method involves reviewing the job analysis and description, determining the criteria, assigning weights, and scoring candidates through interviews and tests. For example, the criterion of teamwork skills may be more important for certain positions, while the criterion of knowledge of specific computer programs may be more important for other positions. Therefore, each criterion is assigned a weight based on its importance, and then candidates are rated and scored accordingly. This approach allows for a fairer and more objective evaluation, although it may not completely eliminate disparate impacts [1].

Statistical models include the compensatory model, the multiple cutoff model, and the multiple hurdle model. The compensatory model allows a high score in one criterion to compensate for a lower score in another criterion. In the multiple cutoff model, candidates must meet a minimum score on all criteria. In the multiple hurdle model, only candidates with high scores above a specified score proceed to the next stages of the selection process. As an example, consider Table 3, depicting a candidate evaluated in a sample selection model that includes criteria with ratings, weights, scores, and interviewers' comments. The rating is multiplied by the weight to obtain the score for each particular criteria, and then the score is summed up [1].

Table 3 Statistical Selection Method Example

Job Criteria	Rating*	Weight**	Score	Comments
Dress	4	1	4	Candidate dressed appropriately.
Personality	2	5	10	Did not seem excited about the job.
Interview questions				
Give an example of a time you showed leadership.	3	3	9	Descriptive but didn’t seem to have experience required.
Give an example of when you had to give bad news to a client.	0	5	0	Has never had to do this.
Tell us how you have worked well in a team	5	4	20	Great example of teamwork given.

Job Criteria	Rating*	Weight**	Score	Comments
Score on cognitive ability test.	78	5	390	Meets minimum required score of 70
			433	
***Rating system of 1-5, with 5 being the highest **Weighting of 1-5, with 5 being the most important**				

Following the example in the table, when using the compensatory model, the ability to give bad news to a client might outweigh the total score because it has a high weight, and if the candidate demonstrated this ability, their total score would be compensated. Using the multiple cutoff model and assuming the candidate is required to have a score of at least 2 points out of 5 on each criteria, the candidate scored low on "bad news to a client," indicating that the candidate is not qualified to fill the position. On the other hand, using the multiple hurdle model and assuming the requirements are scoring 4 points on at least three of the criteria, the candidate in this case scored at least 4 points on three criteria, thus the candidate is qualified to fill the position [1].

2.4.3 Selection Process as a Decision Making Process

Personnel selection is a critical aspect of HRM and plays an essential role in enabling businesses to meet their strategic objectives. It ensures hiring candidates with the appropriate knowledge, skills, experience, and abilities for their job positions. Personnel selection is a complex decision-making process that involves identifying, weighing, and evaluating candidates against a set of criteria and job requirements. The wrong hiring decision can have long-lasting and costly effects, emphasizing the need for effective evaluation and ranking methods in personnel selection. As introduced, traditional personnel selection involves a mix of experimental and statistical techniques, including interviews, work sample tests, and personality tests. Yet, as the field evolves, more advanced decision-making approaches have been introduced, highlighting the transformation of personnel selection into a strategic decision-making process [3] [44] [51] [52] [53].

In this context, a key approach in modern personnel selection is MCDM, which addresses the challenge of evaluating candidates against multiple, often conflicting criteria. Classical MCDM approaches, such as Simple Additive Weighting (SAW), Analytic Hierarchy Process (AHP), Analytic Network Process (ANP), and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), have been widely applied, providing a structured and systematic way to handle complex personnel selection scenarios. Furthermore, classical MCDM methods have been extended to fuzzy environments. Fuzzy MCDM methods, for instance, allows decision-makers to use qualitative evaluations in scenarios where quantitative assessments are challenging [3] [53].

Moreover, the advent of AI has further revolutionized personnel selection with the introduction of expert systems. Pioneered by Edward Albert Feigenbaum in the 1970s [3], expert systems involve transferring human expertise into a computer system, allowing for automated, rule-based decision-making. These systems can be integrated with other methods, such as MCDM and fuzzy reasoning, for enhanced decision-making capabilities. Additionally, fuzzy expert systems are introduced to integrate fuzzy logic and AI, offering sophisticated solutions for personnel selection. Expert systems can evaluate candidates based on a range of criteria, including educational background, work experience, and language skills, demonstrating the flexibility and effectiveness of modern decision-making approaches in personnel selection [3] [53].

Another method used in personnel selection is the Grey system theory. Proposed by Julong Deng in 1982 [3], the Grey theory involves Grey Relational Analysis (GRA), which is particularly effective in dealing with discrete data and making decisions under uncertainty. GRA's capability to process fuzzy and incomplete information makes it a valuable tool in personnel selection [3] [53].

Integrated or hybrid approaches that combine the introduced different decision-making methods are also used in personnel selection for more robust solutions [3] [53]. Other approaches are based on data mining, information retrieval, natural language processing, and machine learning techniques. These approaches often process resumes data to identify and extract patterns representing qualification of applicants and matching it with jobs requirements [54] [55] [56] .

In short, personnel selection has evolved from simple interviews and tests to a comprehensive decision-making process incorporating various advanced methods. This evolution reflects the growing complexity of job markets and the increasing importance of making accurate and strategic hiring decisions in today's competitive business environment.

2.5 Challenges in Recruitment and Selection Process

Competent human resources are vital for any organization, providing difficult-to-replicate competitive advantages and making significant contributions to the organization's development. In particular, Recruitment and selection processes play a crucial role in gaining a competitive advantage and enhancing organizational growth, as they are the primary source of talent acquisition. However, these processes are complex to manage and implement, incorporating several challenges. These challenges can increase business costs, decrease the potential for finding qualified applicants, and thus decline an organization's growth [2].

Emphasizing the importance of competent human resources and effective recruitment and selection processes, Genping Ye [2] discussed the main challenges in recruitment and selection processes and provided corresponding recommendations for organizations to improve their HRM practices. Ye presented the following challenges.

First, many companies have unreasonable and unrealistic job requirements as a result of not linking work standards to actual professional abilities, leading to a lack of applicants and the loss of qualified candidates. Improving communication with all departments in the organization and identifying recruitment reasons can enhance job requirement analysis.

Second, discrimination is a significant issue in recruitment, resulting in the loss of qualified individuals, reputation, and the competitive advantages of a diverse workforce. Discrimination in employment standards can be addressed by following employment laws, eliminating prejudice in job criteria, avoiding discriminatory language in job requirements, ensuring equality in job opportunities, and promoting diversity.

Third, some companies implement ineffective recruitment methods and long recruitment and selection processes, leading to missed opportunities to hire the right individuals and increased time and costs. Therefore, it is essential to assess recruitment methods and select ones that are both cost-effective and suitable for specific jobs and budgets, as well as streamlining recruitment to reduce the timeline, eliminate unnecessary steps, and provide timely feedback. Moreover, saving time and costs can be ideally achieved by using software to manage the recruitment cycle and track applicants, making the process more efficient.

Fourth, firms face several difficulties in attracting candidates, including highly skilled passive candidates, due to poor employer branding, delayed recruitment processes, and negative employee feedback, resulting in low response rates from job seekers and high costs. Consequently, it is vital to enhance the employer brand by building a positive reputation through effective marketing. Furthermore, a quick and honest response to inquiries helps establish a positive employer image and encourages feedback. To attract passive candidates, it is essential to establish and maintain relationships with them, explaining the potential benefits and values of the organization.

Finally, relying only on human decision-making can introduce biases such as favoritism and nepotism in selecting candidates, leading to the hiring of unqualified candidates and increased costs. Another selection problem is a greater focus on professional skills over cultural fit. Using scientific methods and software tools can provide an objective evaluation of candidates and more accurate personality tests, thus helping to select more qualified candidates as well as reducing biases and costs. Likewise, dividing the recruitment responsibilities between the HR and business departments can improve the selection of candidates who are both professionally qualified and culturally aligned. Scientific analysis results and the collective opinion of recruiters should be taken into account before finalizing any hiring decision.

In short, many challenges exist in recruitment and selection processes that can increase business costs and complicate the task of finding qualified candidates. In turn, several steps can be taken to address these challenges, including the effective use of all available human and technological resources to hire the most suitable candidates. Adopting technological methods during the recruitment and selection process can provide objective reference and avoid absolute subjective decisions. Moreover, a large number of applicants can be swiftly filtered to select more qualified applicants for interviews. This enables narrowing the scope of human evaluation, increasing objectivity and the chances of hiring the right candidates, as well as reducing associated costs.

Chapter 3

Erlang Expert Systems

3.1 Erlang Programming Language

To meet Ericsson needs of building telecommunications switches with high concurrency, availability, fault-tolerant and distributed services that can be dynamically upgraded without stopping the running system, Erlang was developed at Ericsson in 80s and since 2000, it has been publicly available [12].

Erlang is a functional programming language primarily designed to develop inexpensive concurrent and transparently distributed as well as high fault-tolerate systems based on the notion of spawning parallel lightweight isolated computational entities called processes that share no memory and interact asynchronously via massages passing resulting in a simplified concurrent programming. These lightweight processes have limited memory overhead allowing to run a vast number of them simultaneously and safely as they share no memory or sate. The processes are managed by Erlang Runtime System (ERTS) or Virtual Machine (VM) independently from the underlay operating system which makes Erlang applications cross-platform [12] [13] [14].

The high availability, reliability or fault-tolerant of Erlang applications is based on the use of distributed machines and isolated processes with asynchronous communication and remote error detection as well as a mechanism of linking and monitoring processes forming a tree structure of workers and supervisors which allows for fixing failures, error-propagation and dynamic code upgrading or replacing without stopping the running system. Supervisors and workers processes need not to be on the same machine and they behave as they were on the same machine. This feature allows for transparent distribution, as such Erlang applications can be developed on a single machine and then deployed on a cluster without major code changes. Furthermore, Erlang employs the philosophy of let it crash opposed to the defensive programming used in most other languages. If a process crashes, other supervising process takes over; provides a fix and restores the system to a safe state. This provides a clean separation between the computation and error correction [12].

As a functional language, Erlang supports immutability, avoiding side-effects, modularity, decomposition, data structures, higher-order functions, (tail)recursions, pattern matching all of which enhance and simplify concurrency. Erlang uses modules as means of defining and structuring functions where functions are first-class entitles that can be passed as arguments and returned from other functions, thus, supporting modularity and decomposition. Furthermore, Erlang is inspired by Prolog logic programming and syntax with the aim of making it syntactically easy to express parallel computations. Basic building blocks of Erlang programs include, immutable variables, atoms to represent constants, tuples to store structs of fixed data, and list to store variable data [12].

Moreover, Erlang provides simple built in concurrent primitives which are: spawn, send (!), and receive. These primitive together can turn a sequential code to a concurrent one. Calls and receives can be done in parallel without blocking or forcing specific order and without concerns of complex issues such as locks, and race conditions. Therefore, parallel waiting time is only that of the longest response as no blocking is involved [12] [13].

Lightweight concurrent processes are a centric concept in Erlang and they are based on the actor model concepts. To solve the problem of modeling parallel communication, Hewitt introduced the actor model in his Planner language work then later with his associates in late 60s and early 70s [57] [58] [59] [60]. Hewitt first used the term actor to define active objects that perform pattern matching in order to initiate actions or activities. Then, the work of Hewitt et al. [59] in 1973 describes a formularization of the actor model to be close to the notion of an agent in Distributed Artificial Intelligence (DAI). Subsequently, the actor model has been evolving by a number of contributions notably the work by Gul Agha and Yonezawa [60] [61] [62] [63] [64] [65] .

The actor model can be described as a lightweight concurrent computational process that asynchronously communicates with other actors via message passing. In this sense, an actor may receive messages queued in its mailbox, independently process messages not necessary in an ordered way, send messages to other know actors including itself, spawn new actors or change its own behavior or state based on the messages it receives [66].

Erlang was the first programming language to implement the actor model as its model of concurrency making it essentially an actor language [60] [67]. Actors in Erlang are called processes and Erlang is arguably the best-known language for actor-based concurrent and distributed programming [67] [68].

Likewise, Erlang distribution comes bundled with Open Telecoms Platform (OTP) which provides a large set of powerful libraries that simplifies developing large applications. It provides a large body of proven working solutions or patterns known as behaviors to build fault-tolerant and reliable systems. The mentioned behaviors include generic servers for implementing client/server architectures, supervisors for building robust systems, publish-subscribe for event and messaging handling and Finite Sate Machine (FSM) for modeling FSMs and relations [14]. As an example, the Erlang/OTP AXD301 ATM switch system at Ericsson is reported with uptime of 99.9999999 percent [13] [69]. OTP also includes powerful subsystems such as a real-time relational database called Mnesia, a multimedia gateway protocol control (Megaco/H.248), as well as documentation building tool (docbuilder). Analysis and testing tools also exist such as Dialyzer for type errors checking, QuickCheek for generating test cases and Wrangler for refactoring [12].

The feasibility of Erlang is industry proven. For instance, Erlang is used to develop the popular WhatsApp instant messaging system [15]. Moreover, leading organizations such as Facebook, Amazon, Yahoo!, T-Mobile, and Motorola are using Erlang in their systems [14]. The Elixir[23] functional programming language is based on Erlang, and it uses the Erlang Virtual Machine (VM) to build scalable systems as well as web and embedded applications. Erlang is also used to develop a number of web frameworks[24], including the Zotonic web framework[25], which uses a fast Erlang-based web server called Cowboy[26]. Several other Erlang applications exist, including RabbitMQ, an advanced message queuing protocol standard; Wing, a 3D graphics modeling framework; and databases such as CouchDB, SimpleDB, and Scalaris [12], as well as Riak27, a distributed NoSQL database.

3.2 Expert Systems

Developed in the realm of AI, expert systems are used for knowledge representation and reasoning with the goal of providing decision making support [29] [70] [71]. An expert system typically consists of a set of facts or assertions about a certain domain known as the Knowledge Base (KB), a set of related production rules known as the Rule Base (RB), and a reasoning program operating on top of the previous two components representing a rule engine [70]. To build a rule engine, therefore, the following high-level requirements need to be identified:

- The KB model encapsulating a set of facts about the domain of interest,
- The RB needed to reason about the knowledge,

[23] https://elixir-lang.org/
[24] https://github.com/ChicagoBoss/ChicagoBoss/wiki/Comparison-of-Erlang-Web-Frameworks
[25] https://zotonic.com/
[26] https://github.com/ninenines/cowboy
[27] https://riak.com/

- The implementation of both the KB and the RB together to represent the inference engine that can reason about the current knowledge concepts, derive new facts and relations, or solve a problem of interest.

3.2.1 Rete Algorithm

Rule engines usually use a repetitive rule-based matching algorithm to iteratively reason about the KB and find matching rules to perform desired actions or solve specific problems. As the size of the problem scales with a large KB and RB, a rule engine becomes computationally expensive [70]. Several researches have been addressing this problem including the creation of the notable Rete algorithm or Rete network by Forgy in 1982 [72]. The Rete algorithm is considered the most efficient pattern matching algorithm and has been used in several popular rule engines such as C-Language Integrated Production System (CLIPS) and Java Expert System Shell (JESS) as well as PS5 and ILOG JRules [36] [73]. To adapt Rete to specific requirements and applications or to enhance performance, many variants and improvements of Rete have been proposed including Treat, Rete/UL, LEAPS, GATOR and others [70] [74]. Figure 4 [70] show a typical rule engine components and execution phases.

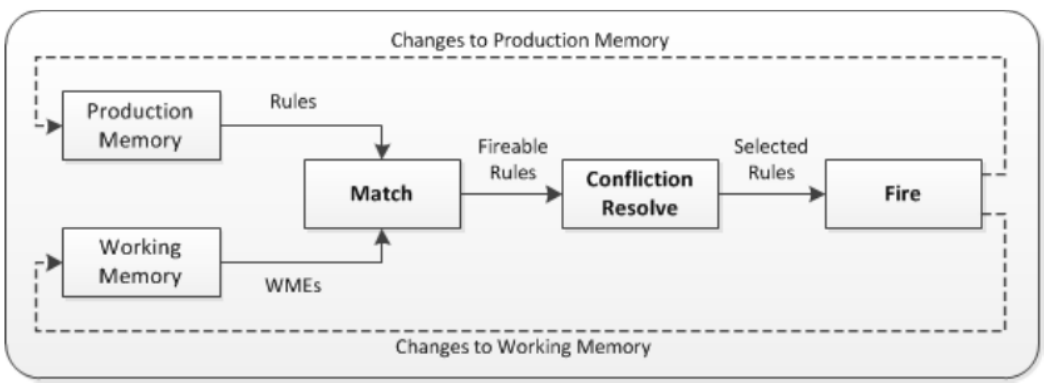

Figure 4 Three-Phase Cycle in Rule Engines

The first two components are Working Memory (WM) and Production Memory (PM) which may change gradually over time or execution. First, the working memory mostly represents the KB or a set of items representing facts. Each item or data is called a Working Memory Element (WME) to represent the current state of the system in the form of a three-tuple consisting of identifier, attribute, and value. Second, the production memory represents a set of productions or rules. Rules are expressed in the form of a condition-action pair, where conditions are in the Left-Hand Side (LHS) and actions are in the Right-Hand Side (RHS). The Inference engine implements the three-phase cycle execution model of match-resolve-fire also known as recognize-act cycle iteratively until a match is found or no more rules can be fired. That is, in each cycle the conditions of each rule are matched against current WMEs. A rule is called instantiation if a set of WMEs matches its conditions. The set of all instantiations is called the conflict set. In sequential executions, one instantiation from the conflict set is selected for firing based on some criteria such as recency, specificity, and priority, whereas in parallel execution, a set of rule instantiations are selected for firing simultaneously according to some correctness constraints such as serializability. If firing an instantiations (executing the actions) led to changes in the current WMEs such as addition, deletion or modification, the matching phase is repeated until there are no more rules to fire or in other words no new instantiations are found [70] [71] [75] .

Erlang is used to develop rule-based and predicate logic-based expert systems. These are discussed in the following sections.

3.3 ERESYE (ERlang Expert SYstem Engine)

Originally known as ERES [16], ERESYE (ERlang Expert SYstem Engine) is an Erlang expert system developed within the framework of eXAT (erlang eXperimental Agent Tool) platform to program agent intelligence or proactive behavior [17] [36]. Based on a modified version of Rete algorithm, ERESYE is essentially used to create rule engines. Its architectural design and runtime mechanism include three main Erlang processes: the Processor, the Rule Scheduler, and the Executor, as shown in Figure 5 [36].

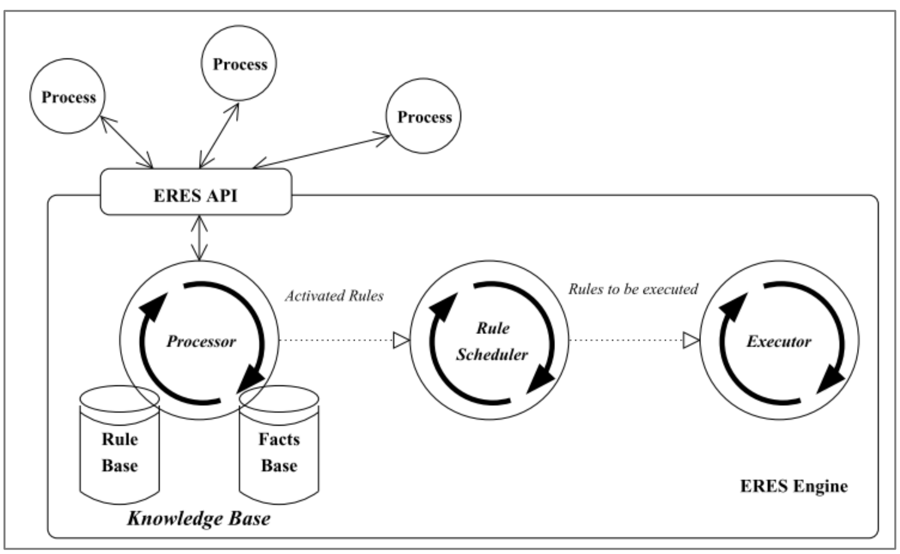

Figure 5 ERESYE Architecture

First, the Processor is an instance of Erlang gen_server behavior. It manages the engine operation and provides an API to interact with other Erlang processes. Second, the Rule Scheduler, which is also an instance of Erlang gen_server behavior, selects rules to be executed based on their priority and according to the engine scheduling policy. Rules can be assigned different priorities to determine their order of execution or they can be scheduled based on their order of activation regardless of their priorities. Third, the Executor process executes the rule sent by the Scheduler process. Rules are executed sequentially, but a rule execution can be detached and started in a separate process especially when adding new rules or in concurrent computations [36].

Based on the previous architecture, ERESYE allows creating, managing, and executing rule-based processing engines. As a typical expert system based on Rete algorithm, an ERESYE engine has a name, a KB, which stores a set of facts representing knowledge of a specific domain application, and an RB , which define inference rules that are used to reason about the knowledge. Facts are coded as Erlang tuples or records allowing to represent KB concepts while rules are expressed as function clauses taking arguments of engine name and facts to match and reason about. Guards can also be used in the function clauses to create additional conditions for the rule to be activated. Moreover, rules are fired based on pattern matching over the associated facts. The body of a function clause implements the actions to be executed if the rule is pattern matched (activated or fired). This function body can contain any Erlang expressions and function calls for KB reasoning and manipulation [36].

ERESYE API provides a set of functions that allow several interaction with the engine such as assert a fact, discard a fact, add a new rule, change rule priority, delete a rule, etc. ERESYE also allows running multiple concurrent engines each with its own KB and RB. Likewise, ERESYE engines are instances of Erlang gen_server and can be defined within Erlang supervision trees, thus, inherent its high fault-tolerance behavior [36].

An improvement for ERESYE regarding its control strategy is proposed in [76]. ERESYE control strategy is handled at a rule level by identifying the rule to activate whenever a change occurs in the knowledge base. However, this introduces a low-level of granularity resulting in decreasing ERESYE performance. Therefore, the improvement in [76] extends ERESYE by introducing an extra layer to perform activations on a sets of rules instead of individual rule to enhance the scalability.

3.3.1 Simple ERESYE Example

The domain of relatives associated with ERESYE documentation[28] represents a simple example that uses concepts of parent, male and female, then, by implementing a rule engine, the concepts of mother, father, sister, brother, grandmother, and grandfather are derived. The base concepts and facts representing this simple domain of relatives with their corresponding Erlang tuple representation are shown in Table 4.

Table 4 ERESYE Relatives Example Base Concepts

#	Concept	Fact / Erlang tuple
1	X is male	{male, X}
2	X is female	{female, X}
3	X is Y's parent	{parent, X, Y}
4	X is Y's mother	{mother, X, Y}
5	X is Y's father	{father, X, Y}
6	X is Y's sister	{sister, X, Y}
7	X is Y's brother	{brother, X, Y}
8	X is Y's grandmother	{grandmother, X, Y}
9	9 X is Y's grandfather	{grandfather, X, Y}

The Concepts 1, 2 and 3 are used as a base to infer the other facts shown in **Error! Reference source not found.**. The rule of deriving the concept of mother can be represented for illustration in if statement as:

$$if\ X\ is\ female\ and\ X\ is\ Y's\ parent\ then\ X\ is\ Y's\ mother$$

In ERESYE facts are stored in the KB of the engine in the form of Erlang tuples or records, while rules are expressed in Erlang function clauses. Each function declares in its head a pattern or a template to match against in order for the rule in the function body to be activated and executed. The following code represents the previous mother rule written as Erlang function in the context of ERESYE with comments in %% explain the code.

[28] https://github.com/gleber/exat/blob/master/doc/Domain_Of_Relatives_Example.pdf

```
%% This % symbol is used to represent an Erlang comment.
%% The mother function in its head it takes arity (argument) of %%
variable named Engine and two tuples as follows:
%% (Engine, {female, X},{parent, X, Y})
%% each tuple consists of Erlang atom data type(a literal),
%% represented by the literals: female and parent.
%% the tuples also include variables: X, and Y.
%% When the mother function is called, it patterns matches
%% against its head: (Engine, {female, X},{parent, X, Y}),
%% if it matches, it executes its body:
%% eresye:assert (Engine, {mother, X, Y}.
%% The imply sign (->)denotes the beginning of the function body.
%% The dot sign (.) denotes the end of the function.

%% The mother function is defined as follows:

mother(Engine, {female, X}, {parent, X, Y}) ->
eresye:assert(Engine, {mother, X, Y}).


%% for instance, if the mother function is called as:
%% mother(my_engine_name, {female, sara}, {parent, sara, tom})
%% the call will match the pattern of the function and:
%% my_engine_name will be bound to the variable Engine,
%% sara will be bound to the variable X,
%% tom will be bound to the variable Y,
%% since it matches, the body: eresye:assert (Engine, {mother, X, Y})
%% is executed resulting in deriving and adding a new fact stating:
%% sara is a mother of tom: {mother, sara, tom}
```

Concepts of father, sister, brother, grandmother and grandfather are similarly written as Erlang functions representing this simple relatives rule engine rules. All of these functions need to be written in an Erlang source file (.erl) known as a module. The following code shows the *relatives.erl* module representing a simple relatives rule engine with its rules and KB directly populated with a set of initial facts or assertions for testing.

```
%% relatives.erl
-module (relatives).
-compile ([export all]).

%% if (X is female) and (X is Y's parent) then (X is Y's mother)
mother (Engine, {female, X}, {parent, X, Y}) ->
      eresye:assert (Engine, {mother, X, Y}).

%% if (X is male) and (X is Y's parent) then (X is Y's father)
father (Engine, {male, X}, {parent, X, Y}) ->
      eresye:assert (Engine, {father, X, Y}).

%% if (Y and Z have the same parent X) and (Z is female) then,
%% (Z is Y's sister)
sister (Engine, {parent, X, Y}, {parent, X, Z}, {female, Z}) when Y =/= Z ->
      eresye:assert (Engine, {sister, Z, Y}).

%% if (Y and Z have the same parent X) and (Z is male) then,
%% (Z is Y's brother)
brother (Engine, {parent, X, Y}, {parent, X, Z}, {male, Z}) when Y =/= Z ->
      eresye:assert (Engine, {brother, Z, Y}).

%% if (X is Y's father)and(Y is Z's parent)then,
%% (X is Z's grandfather)
grandfather (Engine, {father, X, Y}, {parent, Y, Z}) ->
      eresye:assert (Engine, {grandfather, X, Z}).

%% if (X is Y's mother)and(Y is Z's parent)then,
%% (X is Z's grandmother)
grandmother (Engine, {mother, X, Y}, {parent, Y, Z}) ->
      eresye:assert (Engine, {grandmother, X, Z}).

%% start method for this simple relatives rule engine,
start () ->
      eresye:start(relatives),

      lists:foreach (fun (X) ->
            eresye:add_rule (relatives, {?MODULE, X}) end,
            [mother, father, brother, sister,
            grandfather, grandmother]),

      eresye:assert (relatives,[{male, bob},{male, corrado},
                               {male, mark}, {male, caesar},
                               {female, alice},{female, sara},
                               {female, jane}, {female, anna},
                               {parent, jane, bob},
                               {parent, corrado, bob},
                            {parent, jane, mark},
                               {parent, corrado, mark},
                               {parent, jane, alice},
                               {parent, corrado, alice},
                               {parent, bob, caesar},
                               {parent, bob, anna},
                               {parent, sara, casear},
                               {parent, sara, anna}
                               ]
                        ),
ok.
```

The *sister* and *brother* functions include additional guard or condition (*when Y =/= Z*) stating that in order to activate the rule and execute the body of the function, *Y* and *Z* have to be different persons. Moreover, the *start* function creates the *relatives* engine, adds its rules and populates its KB with initial facts by utilizing ERESYE built-in functionality and calling its functions: *start*, *add_rule* and *assert* respectively. From the initial facts, it can be seen that, for example, jane and sara are female and parent. The concepts of mother, father, sister, brother, grandmother, and grandfather are not included in the KB, but based on the defined rules and asserted facts, they are inferred and added to the knowledge base after compiling and running the *relatives* engine. By compiling *relatives.erl* using *c(relatives).* and calling its start function: *relatives:start( ).*, the engine is created and populated, and also the rules are processed resulting in deriving the new facts. The facts in the knowledge base with the new derived facts can be queried using ERESYE function: *get_kb* which returns the last of facts as follows:

```
3> eresye:get_kb(relatives).
[{brother,bob,mark},
 {sister,alice,bob},
 {sister,alice,mark},
 {brother,bob,alice},
 {brother,mark,alice},
 {grandmother,jane,caesar},
 {grandfather,corrado,caesar},
 {grandmother,jane,anna},
 {grandfather,corrado,anna},
 {sister,anna,caesar},
 {brother,caesar,anna},
 {sister,anna,casear},
 {mother,sara,anna},
 {mother,sara,casear},
 {parent,sara,anna},
 {father,bob,anna},
 {parent,sara,casear},
 {father,bob,caesar},
 {parent,bob,anna},
 {father,corrado,alice},
 {parent,bob,caesar},
 {mother,jane,alice},
 {parent,corrado,alice},
 {father,corrado,mark},
 {parent,jane,alice},
 {mother,jane,mark},
 {parent,corrado,...},
 {brother,...},
 {...}|...]
4>
```

ERESYE provides several other functions such as querying specific knowledge. For instance, Alice's brothers only or Alice's and Anna' brothers can be queried using *eresye:query_kb* function as follows:

```
4> eresye:query_kb(relatives, {brother, '_', alice}).
[{brother,bob,alice},{brother,mark,alice}]
5> eresye:query_kb(relatives, {brother, '_', fun (X) -> (X == alice)
or (X == anna) end}).
[{brother,bob,alice},{brother,mark,alice},{brother,caesar,anna}]
6>
```

This ERESYE simple example is used as a prototype case[29] based on this research approach. The example facts and rules are reproduced as RDF ontology and then the rule engine is created. The prototype gives the same results as the original example verifying the research approach.

3.3.2 ERESYE Applications

ERESYE has been mainly used in intelligent agents and robots applications. As mentioned earlier, ERESYE was first developed within eXAT platform to program intelligent aspects of agents. Mapping between ERESYE reasoning and agent interaction is also supported in eXAT which allows for a direct connection between agent message exchanging and intelligence. This eXAT feature of agent aspects integration provides the implementation of "a true rational" agents [17]. Moreover, for agents communication, eXAT supports Foundation for Intelligent Physical Agents (FIPA) specifications and also user-defined semantics or ontologies expressing concepts in a form of classes with hierarchies that are integrated and used in agent behaviors, agent messaging and ERESYE [19]. ERESYE supports defining custom ontologies and provides ontology compiler tool which allows parsing the ontology file and generating relevant Erlang code. These ontologies use Object-oriented programming (OOP) class hierarchies similar concepts based on which an equivalent Erlang include file (.hrl) is generated containing Erlang records corresponding to the provided classes. An Erlang source file (.erl) is also generated which includes functions to manipulate the ontology and maintain classes relations[19] [36]. An example of the described ontology handling is shown in Figure 6 [36].

[29] https://github.com/MiloudEloumri/match/tree/main/apps/semantic_relatives

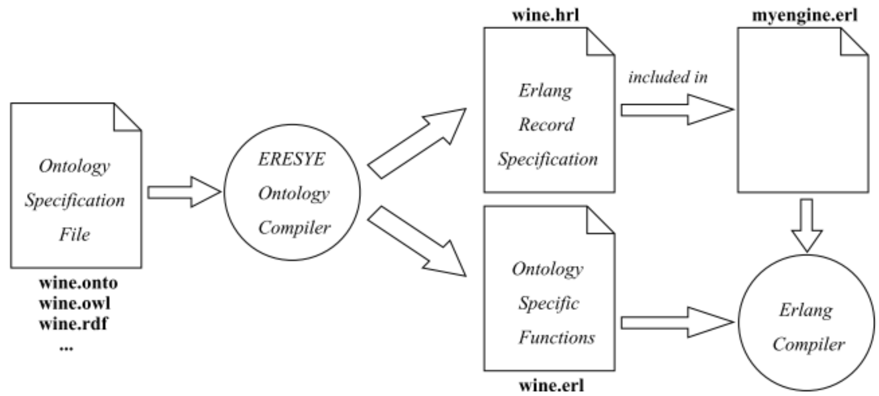

Figure 6 ERESYE Ontology Handling

The ontology compiler tool generates an Erlang source file (parser file) called codec which is used internally by eXAT to perform automatic Erlang/FIPA translation [17] [19]. The ERESYE ontology support requires particular care specially in terms of rules definitions and functions clauses pattern matching duo to the mismatching between OOP concepts and Erlang [17]. This custom ontology handling was removed in the enhanced version of ERESYE called SERESYE , which is discussed in the next section. The authors of ERESYE [19] [36] stated that the support for standards such as Web Ontology Language (OWL) ontologies is planned in future releases by means of translating OWL files or by means of developing Protégé (popular ontology editor) plugin for ERESYE. Then, these OWL ontologies are handled by the ontology complier tool. Additionally, ERESYE can be used independently from eXAT and also apart from rule processing, for example, to perform coordination among parallel processes using Linda coordination language concepts [36].

Based on eXAT, Peregud at al. [34] developed an intelligent resource monitoring and management system mainly for grid-based distributed systems but applicable to cloud environments and standard LAN infrastructures. The system aims to support network managers by employ intelligent agent as means of not only monitoring and reporting performance metrics and detecting certain network failures but also inferring knowledge about the state of the system from collected data and acting upon it without the need for manual configuration. The authors found that ERESYE provides an effective reasoning approach for real-time and dynamically changing environment. It is especially good to use in central decision-making systems. Furthermore, it is found that the integration between eXAT and ERESYE is good, but the use of the system as a whole is complicated such that, it is easier to use ERESYE alone directly through its API.

ERESYE is used to implement a control and intelligent strategy for an autonomous mobile robot called Caesar [77]. Unlike robots that are used in automated manufacturing plants to perform repetitive tasks generally in a static environment, autonomous mobile robots operate in a dynamically changing environment to satisfy their specific design goal. This autonomous behavior bound to environmental changes requires a software structure and the use of proper AI algorithms to allow the robot to perceive and reason about changes in the environment, then, adapt its behavior accordingly and execute actions towards achieving its goal [77]. An Erlang framework for programming autonomous mobile robots is introduced in [77]. The framework enables designing and implementing control and intelligent strategy that can be adapted to specific robotic applications. Its design is based on a layered software architecture which properly separates robotics hardware concerns from its control and intelligent strategy functionalities in order to simplify the system and to allow for code reuse. In addition to many other layers, the intelligence layer of the system is based on ERESYE which provides the implementation of the robot's control strategy by means of specifying production rules that reason over the robot's facts and the changes in the environment. Moreover, reusing of software components for different types of mobile robotic applications and also when changing some mechanical components is made easy and possible by the layered architecture of the framework and also by the use of Erlang OTP behaviors. Such that, the entire framework is organized as an OTP application and is run as a set of Erlang processes linked in a supervision tree to allow for high fault-tolerance.

3.4 SERESYE (Swarm oriented ERlang Expert SYstem Engine)

SERESYE[30] (Swarm oriented ERlang Expert SYstem Engine) is an enhanced version of ERESYE. It includes ERESYE core features with the goal of designing it to be small, self-contained, mobile, and scalable that allows organizing rules to propagate from engine to engine. SERESYE is integrated with the currently available and modified versions of eXAT[31] and eXAT+[32].

3.4.1 SERESYE Applications

Similar to ERESYE, SERESYE has been used within intelligent agents framework. The original version of eXAT platform does not provide the functionality of agent migration across nodes. The support for agent migration was addressed in [35] and added to eXAT. The modified version of eXAT is referred to as eXAT+. The features and improvements provided by eXAT+ are agent

[30] https://github.com/afiniate/seresye
[31] https://github.com/gleber/exat
[32] https://github.com/michalwski/exat

migration, agent registration on different platforms and agent creation performance boosting. The eXAT+ uses SERESYE instead of ERESYE for better performance.

3.4.2 Updating SERESYE

To align SERESYE with recent versions of Erlang and its associated tools, such as Rebar3, the thesis author undertook several enhancements on Gleber SERESYE[33] version. These enhancements included updating configuration files, incorporating 'spec' and 'doc' annotations in various modules for better documentation and type specifications, and updating its supervisor module, as well as renaming and updating its server module to be consistent with Erlang's Open Telecom Platform (OTP) standards. As a result, this thesis utilizes the version of SERESY[34] that has been updated and maintained by the thesis author, ensuring compatibility with the latest versions of Erlang, Rebar3, and other dependent packages and libraries.

3.5 RUNES: An Erlang Expert System

To exploit the resources of cloud computing in terms of processing big data, Wang et al. developed Erlang-based distributed and scalable rule engine called RUNES[35] [70]. RUNES is used to match big data with a large number of rules. The authors then introduced an improved version of RUNES called RUNES II [78].

The matching algorithm used in RUNES II is based on an improved version of the Rete algorithm [78]. The Rete network is implemented as groups of interconnected Erlang processes that communicate in parallel through message passing. Moreover, RUNES II include resource cost models and an algorithm of allocation to allow for high performance and for efficient use of resources on the cloud. First, nodes in Rete network represent Erlang processes that can be distributed in different virtual machines. Second, the resource cost models are used to measure the virtual machine usage of resources such as memory and response time in order to optimize the quantity of the used virtual machines. Third, the algorithm of allocation is used to improve the efficiency of rule matching by controlling the number of passed messages and balancing the load among different computer nodes. Each virtual machine represents an independent agent the runs a full image of RUNES II rule engine with all its related processes that are supervised and controlled

---

[33] https://github.com/gleber/seresye
[34] https://github.com/MiloudEloumri/seresye ;
https://github.com/MiloudEloumri/match/tree/main/apps/seresye

[35] https://github.com/b7ack42/runes

by the agent itself. The number of initial virtual machines are determined based on the scale of rules and facts and used to build a cluster. The agent in every virtual machine uses the allocation algorithm to create Rete processes based on the rule set. Then, agents of any virtual machine exchange facts to match and process rules. RUNES II is deployed in the USTC Cloud platform which provides on demand virtualization and resources. In this regard, RUNES II Erlang program is deployed in virtual machines using a shell script which starts new virtual machines, initializes Erlang environment, constructs control processes, and adds more new virtual machines dynamically if the rules increased in the cluster to improve the performance.

RUNES II use the same design and implementation as RUNES but provides improvements in terms of adding process allocation algorithm and resource cost models mentioned previously. In this context, the design of RUNES is based on (1) mapping the Rete Algorithm to the message-passing model, (2) distributing the RUNES in cloud environments, (3) allocation of rules and facts, and (4) optimization. Figure 7 shows the main modules used in RUNES system [70].

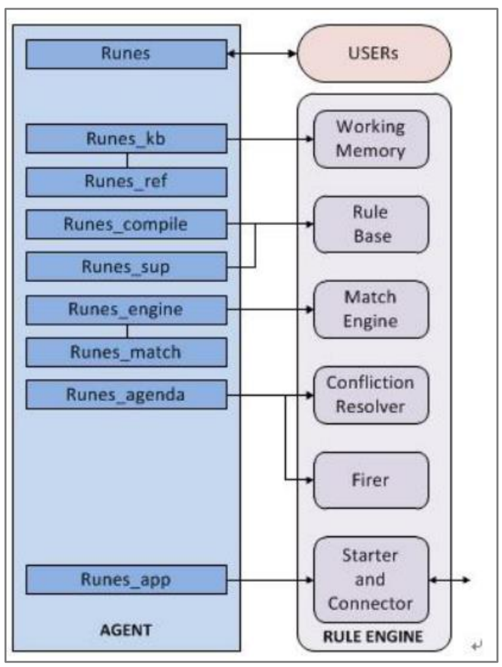

Figure 7 RUNES Architecture

- The Runes module provides users with an interface to add or delete facts and rules as well as getting various results.- The Runes_app module starts agents and their communication.

- The Runes_sup module starts one supervising process for the started agent making the agent as its working processes.

- The Runes_compile compiles rules into the rete network and links them to the corresponding working processes (agents).

- The Runes_agenda module collects information about the whole system and resolves the conflict set and activates selected rules.

- The Runes_engine module runs the match algorithm and controls data and activation flows to match facts against rules with messages' creating transferring among rete-node processes.

- The Runes_match module implements concrete test algorithm used in the match phrase.

- The Runes_kb module manages the working memory (facts) and maintains the current state of the system using the built-in distributed database of Erlang.

- The Runes_ref module provides a unique global reference for the system resources.

3.6 Erlog: Prolog Interpreter in Erlang

Unlike the previously introduced Erlang expert systems which are based on rule production and Rete algorithm, Erlog[36] is based on predicate logic of Prolog logical programming language. Erlog is a Prolog interpreter written in Erlang and integrated with its runtime system. It represents a subset of the Prolog standard and it includes an Erlog shell (REPL). The purpose of Erlog is to provide a logic-based inference engine that can be used in Erlang applications for reasoning and inferring knowledge [79].

3.7 Expert Systems Applications in Personnel Selection

As introduced in the previous sections, Erlang-based expert systems are mainly used in intelligent agent applications and autonomous robots. To the best knowledge of this research, this is the first study that uses Erlang expert systems in personnel selection problem. However, several other expert systems approaches are used in addressing this problem.

The primary objective of expert systems is to facilitate decision-making; therefore, it is logical that several expert systems are particularly developed for use in HRM to assist managers in critical decision-making. HRM domains such as planning, job analysis, recruitment, selection, performance evaluation, compensation, training, and labor–management relations are the most suitable for the development of expert systems [80] [81].

[36] https://github.com/rvirding/erlog

Byun and Suh [80] provided comprehensive guidelines to human resources managers on the application of expert systems in various HRM domains, emphasizing the importance of these systems in improving several HRM activities. Moreover, they introduced a wheel model representing the development of Human Resource Management Expert System (HRMES) with semantic networks for knowledge representation. The HRM activities where expert systems can be effectively implemented include planning, job analysis, recruitment, selection, performance evaluation, training, labor-management relations, and compensation. The study emphasized the critical role of knowledge representation in the development stages of expert systems. The authors concluded that expert systems offer more effective and efficient support for complex managerial decision-making. Furthermore, expert systems provide a competitive advantage to organizations and assist in operational, tactical, and top-level management decisions.

Using Prolog, Hooper et al. [81] developed a rule-based expert system called BOARDEX for the selection of officer personnel in the U.S. army specifically for admission to the Command and General Staff College (CGSC). BOARDEX was designed to enhance the personnel selection process by handling the initial screening of officer records based on various criteria such as grade, military and civilian education levels, height, weight, assignment history, and Officer Efficiency Report (OER) evaluations. The study demonstrated that BOARDEX's decisions were not statistically different from the decisions made by human experts leading to saving time and resources needed for the personnel selection process while maintaining or improving the quality of results. This study provided a thorough review of early research on the use of expert systems in the military and HRM. For instance, several HRM-related expert systems were highlighted, including: (1) Service Selection Advisor (SSE) developed by Virginia Polytechnic Institute and State University to assist in career field choices for senior midshipmen in Naval ROTC programs; (2) Organizational Consultant designed to assist in organizational structure decisions; (3) Resumix developed for fast evaluation of resumes and job applications, and (4) EXPER developed as part of an overall HRM system and focused on job placement by evaluating and matching employees skills with suitable job roles. These examples show various efforts and applications of expert systems in HRM that aim to enhance decision-making and manage large volumes of data effectively. While expert systems offer such advantages including increased productivity, consistent performance, and institutionalized expertise, as well as valuable support in various HRM tasks and decision-making, they are not without limitations [81].

Using CLIPS[37] (C-based programming language for building expert systems), Mehrabad and Brojeny [82] implemented an expert system for personnel selection and appointment. The objectives of the developed expert system include: (1) enabling the selection of suitable job applicants based on job and organizational requirements, (2) supporting the appointment of suitable jobs for applicants, considering job classification and organizational needs, and (3) determining appropriate salaries and benefits for applicants based on their qualifications. The developed Expert system uses a rule base representing criteria for jobs and candidates specification. These criteria were extracted from job analysis and expert judgments. Moreover, the Multidimensional Scaling Model (MDS) was used to extract hidden factors for successful job performance. The system was tested in an R&D organization with various departments, where it was used to make decisions about suitable R&D jobs, required job training, workplace assignment, and salary determination. The research emphasized the value of expert systems in supporting various personnel operations and highlighted their flexibility in handling unstructured data, engineering knowledge base, and providing various decision-making support through explanation and recommendation.

In a similar context of developing rule-based expert systems specifically for personnel selection, some approaches integrate other theories such as data mining and set theory. Chien and Chen [83] investigated the issues of high turnover rates and recruiting the right talents in high-tech companies. They found that developing effective personnel selection mechanism can solve these problems. Accordingly, the authors developed a data mining framework based on decision tree and association rules to extract rules for personnel selection. To generate the decision rules, the framework identifies the relations between personnel attributes and work behaviors, including performance and retention. Moreover, the framework is empirically tested particularly for engineers roles in a large semiconductor company located in Taiwan.

Another personnel selection based on expert system and Rough Set Theory (RST) was proposed by Akhlaghi [84]. Akhlaghi explored the use of RST as a mathematical approach to address the challenges of personnel selection, especially when dealing with qualitative data and fuzzy criteria such as decision-making ability, adaptability, ambition, and self-organization. The approach used Rough Set Exploration System (RSES[38]) to analyze data, generate decision rules, and identify the matched attributes for each candidate. Furthermore, the system was tested in an IT company in Iran, where ten attributes relevant to IT professionals were identified and weighted by experts from

---

[37] https://www.clipsrules.net/
[38] https://www.mimuw.edu.pl/~szczuka/rses/start.html

the industry. These attributes were then used to score and rank the company's personnel. A total of 1271 candidate rules were processed, and then the rules with more matched attributes are considered the process outcome, resulting in 4 matched candidates.

In the context of integrating MCDM methods and expert systems concepts in personnel selection, Jereb et al. [85] presented an approach to personnel selection integrating MCDM with expert systems using DEXi[39] tool. DeXi is a multi-attribute decision-making program developed in collaboration between Josef Stefan Institute and University of Maribor, especially for the interactive development of qualitative multi-attribute decision models and the evaluation of options. DeXi supports the decision-making process by constructing and evaluating a knowledge base including model and attributes structuring, decision rules checking, options analyzing, and results explaining and reporting. Jereb et al. approach is based on the explicit expression of qualitative decision knowledge in a hierarchical tree of attributes and decision rules. This method differs from quantitative models by using qualitative attributes (expressed in words rather than numbers) and discrete functions defined in table forms for decision-making. The methodology of personnel selection in this study involved several stages, including: (1) problem identification: identifying the specific properties of the personnel selection problem and defining goals and restrictions, (2) project setup: involving problem owners, experts, decision analysts, project team and choosing the appropriate modeling methodology, (3) modeling: developing a qualitative multi-attribute model through collaboration within the problem-solving team, involving brainstorming, discussions, and interviews, (4) option identification: collecting and describing options (candidates) for the selection process, (5) option evaluation and analysis: utilizing DEXi for option evaluation and analysis, including what-if and sensitivity analysis, and identifying key advantages and disadvantages of options, (6) decision-making: based on the collected evidence, making the final decision or establishing a preference order of options, (7) deployment and implementation: documenting and communicating the decision in a transparent manner and implementing the decision effectively. The practical application of this methodology was illustrated through the selection of a top manager in a pharmaceutical company. This study shows that MCDM concepts, when integrated with expert systems principles as in DEXi, can be highly useful in personnel selection.

[39] https://kt.ijs.si/MarkoBohanec/dexi.html

Other expert systems for personnel selection combine Artificial Neural Networks (ANNs), Genetic Algorithm (GA), intelligent agents and fuzzy logic such as in [86] [87] [88] [89] [90] [91].

Key limitations of expert systems in HRM applications include the use of incorrect knowledge, challenges in knowledge acquisition, difficulties in knowledge representation, lack of learning and handling complex cognitive tasks, as well as the human fear of computers overtaking the human aspect of decision-making. Some of these limitations can be addressed through advanced programming and regular system maintenance. In generic, it is important for HRM to use expert systems as a support tool rather than as a standalone decision-maker and ensure final decisions are made by humans [81].

.

Chapter 4

The Semantic Web

4.1 Historical Context: From Mesh to World Wide Web

In 1989, the problem of managing information and preventing their lose at CERN (the European organization for nuclear research) inspired Tim Berners-Lee (the inventor of the worldwide web) to propose a solution based on a distributed hypertext system naming it Mesh [92]. In 1990[40], Berners-Lee renamed his proposal from Mesh to World Wide Web (WWW) paving the way to the current well-known web. Since then, enormous number of web pages containing huge information in the form of linked documents and media in various formats have been exponentially grown on the web turning it into a massive easily accessible global medium of information. Moreover, information retrieval has been supported through several search engines that help in fetching relevant information. However, the exponential growth of the web has made it increasingly difficult to efficiently manage, index and intelligently process this massive information by the search engines in terms of both precision and recall using traditional ranking and searching algorithms [93]. In essence, the information on the web is meaningful to human without explicit and formal semantics or meaning for programs or agents to intelligently process the information.

4.2 The Inception of the Semantic Web

The previously mentioned problems associated with web information management, search, and intelligent processing led Tim Berners-Lee et al. in 2001 to propose an extension to the web calling it the semantic web [37] (technical proposal of semantic web development by Tim Berners-Lee appeared earlier on February 4, 2000 [94]). Moreover, the semantic web concept was essentially pointed out by Berners- Lee much earlier in 1994 at the first World Wide Web Conference but it showed slow progress until later where the actual development has started [41].

As a web extension, the vision of the semantic web is to give a well-defined or formal meaning (semantic) to online information enabling machines to better process and understand the data in order to perform or automate various tasks and allow for more enhanced cooperation and information exchange [37]. Initially, Berners-Lee et al. [37] discussed the semantic web in terms of four main concepts: expressing meaning, knowledge representation and reasoning, ontologies

---

[40] https://www.w3.org/History/1989/proposal.html

and putting explicit emphasis on intelligent agents that can process semantic data allowing to make adaptable plans for users and for information searching, exploring and exchange. *"The Semantic Web is not a separate Web but an extension of the current one, in which information is given a well-defined meaning, better enabling computers and people to work in cooperation. Berners-Lee, 2001"* [37].

The concept of the semantic web as it evolves, it introduces more other concepts and terms. For instance, it is observed that there is an unclear conceptual relationship among common terms of semantic web, web of data and linked data. These terms are occasionally used interchangeably, treated differently, or viewed as overlapping concepts, but some complex inter-relational among the terms exists [93].

Based on the discussion presented in [93], the semantic web refers to several concepts including: making the web a global decentralized database (web of data) with its related applications, set of specifications (standards, languages, vocabularies, protocols) along with principles and practices related to open linked data to frame the whole concept. In this regard, the term web of data refers more to the act of adding semantic data to the web allowing to perform database similar queries and transfer the current web from the web of passive linked documents to the web of actionable semantically interrelated data or information that are machine-processable. This indicates that the term web of data can be treated as an objective of the semantic web [93]. Similarly, the third concept of linked data is within the semantic web frame referring to a set of principles and rules for publishing, sharing and exchanging data among different sources on the web. In short, the linked data concept is an application of semantic web technologies together laying the foundations for the web of data as a global database space. This discussion can be illustrated visually as shown in Figure 8 The Concept of Semantic Web [93].

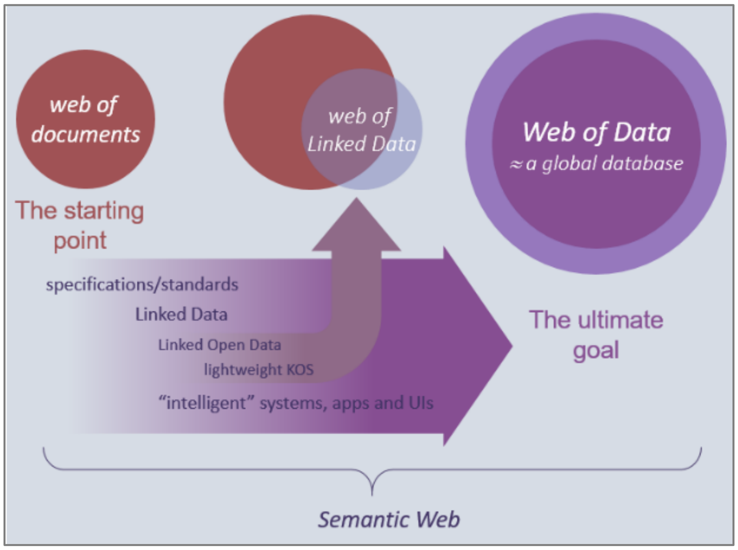

Figure 8 The Concept of Semantic Web

Several organizations have been supporting the web of data objective by adopting linked data practices and publishing their data based on semantic web technologies . The Linked Open Data (LOD) Cloud[41] shows a grown number of datasets that have been published in the linked data format.

4.3 Semantic Web Technology

The semantic web has evolved over time resulting in introducing several technologies and standards. Figure 9[42] groups most used figures that show how the semantic web technology had evolved since its 2000 technical proposal [94] to 2005, whereas Figure 10 shows a recent version of semantic web technology that include more standards such as a variety of RDF document types, JSON-LD, and others.

[41] https://lod-cloud.net/
[42] https://medium.com/openlink-software-blog/semantic-web-layer-cake-tweak-explained-6ba5c6ac3fab

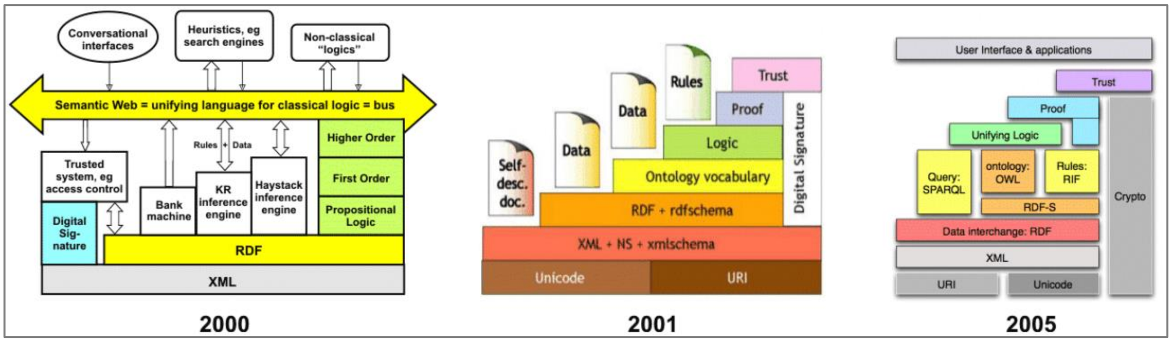

Figure 9 Semantic Web technology 2000-2005

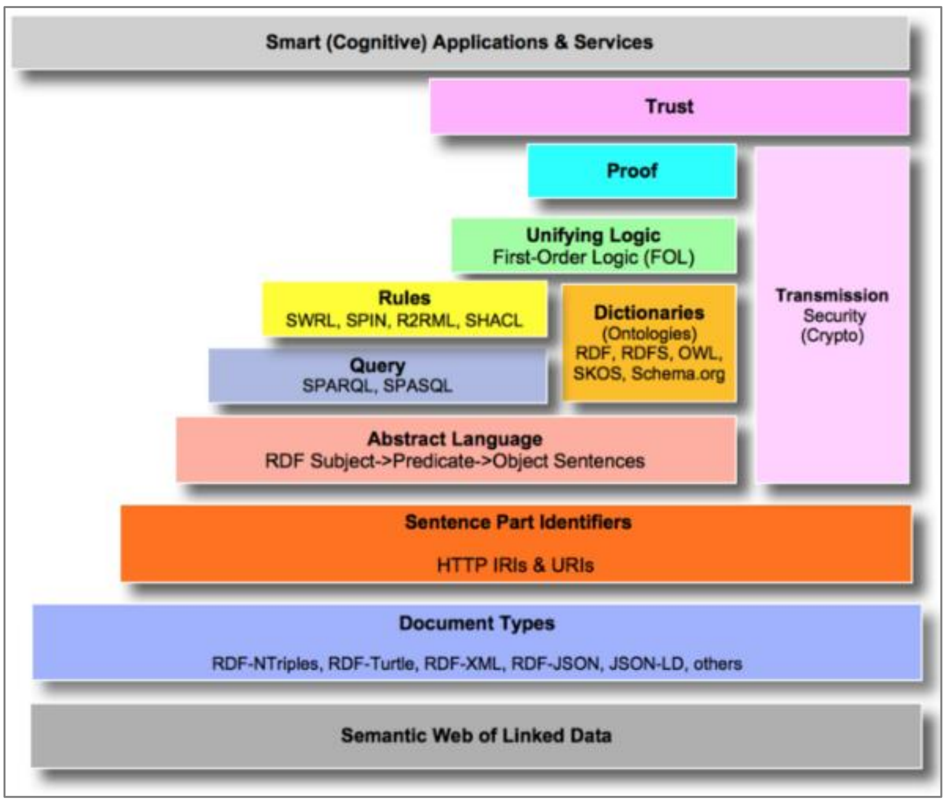

Figure 10 Semantic Web technology 2017

Another interesting semantic web technology visual representation is based on isometric projection as shown in Figure 11 [93].

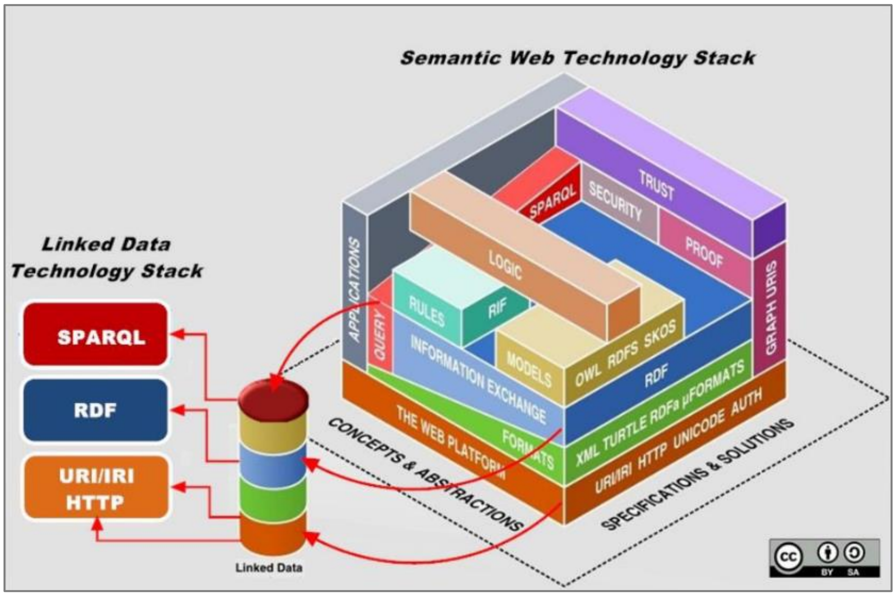

Figure 11 Semantic Web technologies stack

The previous semantic web figures show the layers of the technology with higher-level components using the syntax and semantics of lower levels. A brief description of main semantic web technologies is presented in Table 5 based on information extracted from [38] and [39].

Table 5 Description of Main Semantic Web Technologies

Item	Description
Uniform Resource Identifier (URI)	A unique identification of resources
Resource Description Framework (RDF)	A standard data model to represent knowledge or information in the semantic web. RDF data takes a triple form of subject, predicate, and object. URIs are used in RDF to identify resources. The predicate is a URI representing the relationship between the subject and the object. The subject can be a URI, or a blank node and the object is an URI, blank node, or a literal type. An RDF triple is also known as RDF statement. An RDF document represents a directed graph model. The nodes in the RDF graph can be a resource (URI) or a literal (subject and object) whereas the edges represent the relationship (predicate). Several notations or syntax of saving or serializing RDF data exist including RDF-XML, RDF-Turtle (Terse RDF Triple Language), RDF-N-Triples, N-Quads, JSON-LD (JavaScript Object Notation for Linked Data), etc.
RDF Schema (RDFS)	A simple ontology language describing vocabularies or concepts(resources) as a metadata model representing a structured formal RDF model. RDFS allows defining and instantiating class, subclasses as well as properties and sub-properties with their domain and range restrictions to define an RDF model consisting of a hierarchical relationship among described resources. Attributes with XSD (XML Schema Definition) datatypes can also be used in RDFS. RDFS is semantically simple as it does not include expressivity of features such as exclusion or negation

Item	Description
	which limits its logical inference. OWL (see next) extends RDFS and offers more powerful logical deduction.
Web Ontology Language (OWL)	Built on top of RDFS, OWL is the current prominent web ontology language standard offering powerful semantic and logic expressivity. In addition to features supported in RDFS, OWL supports many other features such as defining complex classes from simple ones which allows for class exclusion. Some variants of OWL which differ in their expressivity level exist such as OWL Full and OWL DL (Description Logic). OWL Full has the expressiveness of the First Order Logic (FOL) which is undecidable whereas OWL DL is considered the main means of building ontologies as it is based on the fragments of FOL known as Description Logics (DLs) which is decidable.
Rule Languages	Rules for logic programming and knowledge reasoning based on DLs are supported in web ontologies and integrated with OWL. Such rules are provided for example by the Semantic Web Rule Language (SWRL). Exchanging rules among web rule systems is also supported through Rule Interchange Format (RIF) standard. These integrated rule systems complement RDFS and OWL allowing for enhanced reasoning.
The Web Service Modeling Language (WSML)	WSML, as a standard ontology language, supports annotating semantic web services aspects based on formal logic.
Simple Knowledge Organization System (SKOS)	SKOS focuses on supporting controlled semantic vocabularies used with ontologies and RDF standards.
Protocol and RDF Query Language (SPARQL)	A standard query language for RDF. It allows to process RDF graph and retrieve various data based on node navigation and graph pattern matching. SPARQL uses SQL like queries (SELECT, FROM, WHERE, etc.) with prefixed namespaces to find a sub-graph in the RDF graph that matches some query. SPARQL is not only a query language, but also a protocol layer to extract RDF data from http requests. SPARQL quires are performed on RDF datasets (databases) known as RDF stores. There exist different RDF stores which can be categorized as RFD native and non-native (see Figure 12 [39]). Native RDF stores (e.g., RDF-3X, GraphDB, Blazegraph, AllegroGraph) are entirely built based on RDF data model whereas non-native RDF stores (e.g., Jean, 3store, Redis, CouchDB) add an RDF interface to other excising database systems. The mechanism of these RDF stores differs in terms of handling data size, performance, inference support and other factors.

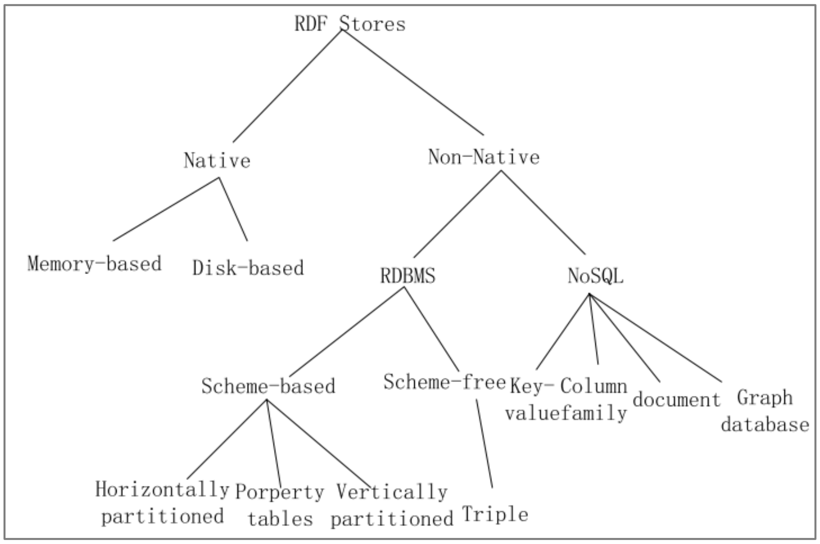

Figure 12 RDF Store Taxonomy

4.3.1 Semantic Web Ontology Concepts

The origin of the term ontology with its basic concepts is dated back to the ancient Greek philosophy where it was used to denote the study of existence including categorizing and relating objects or things [38]. It tries to answer questions such as ''what kinds of things are there?' This basic ontology concept was then transferred and adapted into knowledge representation and Computer Science to develop conceptual yet computational knowledge models of an application domain. Several definitions of ontology based on different contexts exist. A concise definition of what an ontology is in terms of the semantic web is given as: *"an ontology is a formal explicit specification of a shared conceptualization of a domain of interest."* [38]. This definition proposes some important aspects of the ontology, as follows:

Formal- An ontology must be well-defined and machine-processable by means of expressing it in a knowledge representation language (such as OWL) based on formal semantics and logic to ensure that the specifications of the domain knowledge captured in the ontology are interpreted semantically and logically correctly.

Explicit- Ontology concepts must be explicitly stated or defined otherwise the concepts are not machine-processable.

Shared- To be shared, there must be a community agreement about ontology concepts of a certain domain.

Conceptualization- An ontology is captured as a general abstract model identifying concepts with their relations.

Domain– An ontology specifies knowledge of a particular domain of interest where limiting the scope of the domain allows for capturing more accurate details.

4.3.1.1 Ontologies Applications

Ontologies are not only the enabling technology of the semantic web, but they also have become an important subject in Computer Science allowing knowledge models to be processed by various systems. They provide means of formal explicit knowledge representation and semantic vocabulary to annotate online information which allows for sharing machine-processable data. Providing such formal and explicit domain knowledge allows systems to interact with it at runtime and to share it with other systems. Moreover, ontologies combine concepts from symbolic knowledge representation in AI, formal logic, automated reasoning, and conceptual modeling in Software Engineering empowering more features and functionality [38]. Modeling knowledge using ontology specification enables expressing facts and concepts easily and organizing them with meaningful structures and relationships. Accordingly, ontology technology has shown increasing work and research such as in formalizing and standardizing knowledge representation, ontology languages, methodologies, tools, linked data, semantic search and information integration, Service-Oriented Architecture (SOA) as well as in intelligent agent systems [33] [38] [40] [41].

OWL as the main sematic web ontology language provides automatic reasoning to deduce implicit knowledge represented in the underlying ontology and perform various checks such as concepts/objects definitions consistency and completion, subsumption testing and classification of new instances and concepts. However, OWL underlying DL allows only to express static aspects of the domain and not expressing state transitions. Therefore, it is not possible to use OWL to model dynamic interactions with task execution which is required when developing intelligent systems. An attempt to address this problem has been proposed and called OWL-POLAR. OWL-POLAR supports OWL-based knowledge representation and reasoning based on a concept of policies. Policies or norms are machine-processable declarations of constraints and rules allowing to develop distributed intelligent systems [33]. Furthermore, A Knowledge-Based System (KBS) engineering approach based on semantic web technologies is proposed in [95] [96]. The KBS resembles expert system concepts and it consists of two main modules: knowledge base and control system. First, the knowledge base incorporates three main parts: factual, class and rule knowledge. The factual knowledge represents case-specific declarative knowledge describing facts of the system. Class knowledge represents domain-specific declarative knowledge stating the domain knowledge requirements and relations based on semantic terms. The rule of knowledge represents declarative and procedural knowledge that allows to produce, change, or remove facts based on the available factual and class knowledge. Second, the control system includes two main components: interface to present the system to the user for various operations and inference engine for problem-solving strategies. The interface comprises three parts: interviewer, explanation and knowledge acquisition which are used to interact with the user in a form of getting input and generating explanation and output. The inference engine operates independently from the knowledge base and it applies rule knowledge to class and factual knowledge in order to solve a problem and to (re)write new factual knowledge or to search on the factual knowledge. Figure 13 [89] shows the described system.

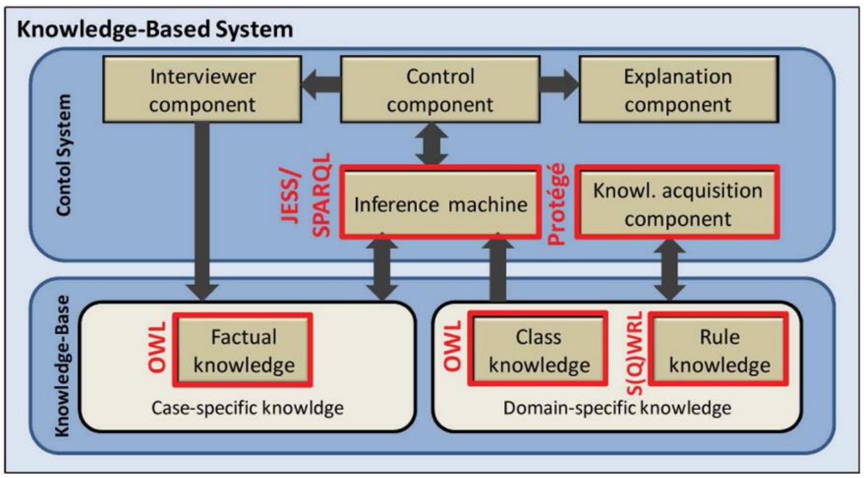

Figure 13 Knowledge-Based System Using Semantic Web Technologies

As shown in Figure 13, OWL is used to represent class knowledge and factual knowledge to allow for machine formal semantic interpretation. Moreover, rule knowledge is represented by SWRL-Rules (Semantic Web Rule Language) where the search queries are encoded in Semantic Query-Enhanced Web Rule Language (SQWRL) which is the query language of SWRL. Likewise, Jess, a Java-based rule engine is used for inference and reasoning. The authors' justifications of their use of Jess include its support for OWL and SWRL/SQWRL and its backward and forward reasoning capabilities. In addition to Jess, the system allows the use of SPARQL for OWL ontologies searching providing similar functionality of SQL in traditional database.

4.4 Semantic Web Toolkit for Erlang Applications

There is a few work in terms of supporting sematic web data processing in Erlang applications. This work is mainly available as online code packages such as Lagra[43] and Semantic Web Toolkit for Erlang Applications[44]. Other similar packages written in Elixir (Erlang-based functional programming language aimed for web development) are also available including RDF.ex[45]. This thesis uses Semantic Web Toolkit for Erlang Applications to integrate the processing of semantic web data with the rule processing of SERESYE.

Semantic Web Toolkit for Erlang Applications is an Erlang library that interprets semantic web data as a collection of ground facts or knowledge statements. It defines Erlang native formats for the knowledge statements and their serialization formats. It also provides rules for data type mapping between semantic web and Erlang applications, resolving the complexity of Internationalized Resource Identifier (IRI) identity and providing built in database of semantic name-spaces. The toolkit supports semantic codecs for N-Triple, JSON-LD, and pure JSON formats.

Semantic Web Toolkit for Erlang Applications is used in this research to decode BTM jobs-applicants ontology in Erlang terms representing the KB of the developed SERESYE job-applicant matching and evaluation rule engine.

Using both SERESYE and Semantic Web Toolkit for Erlang Applications can help in creating powerful, scalable, and flexible rule engine that leverages the benefits of both worlds, Erlang, and semantic web technologies. Erlang is known for its concurrency, fault tolerance and scalability features, which enable creating fast, reliable, and scalable applications. Therefore, using SERESYE and Semantic Web Toolkit for Erlang Applications, both built on Erlang enables creating a rule-based expert system that is both rapid and effective. Moreover, utilizing semantic web technologies enables creating a rich KB resulting in more accurate reasoning capabilities and efficient decision-making process as well as improved data sharing between different systems.

[43] https://github.com/darkling/lagra
[44] https://github.com/fogfish/semantic
[45] https://github.com/rdf-elixir/rdf-ex

Chapter 5

Multi-Criteria Decision Making

5.1 The Foundations of Decision-Making in Complex Scenarios

In situations where decisions must be made (e.g., selecting a candidate for a job position), it is feasible to briefly discuss the complexity of the decision-making process. Regarded as a complex mental process, decision-making is a problem-solving mechanism that involves determining a desired result considering various related factors and alternatives. Such a decision-making process can be rational or irrational. Moreover, it may make implicit or explicit assumptions that are influenced by many factors, such as physiological, biological, cultural, and social factors. These factors combined with the authority and risk levels may further deepen the complexity of a decision-making process [44].

Decision-making scenarios are commonly found and dealt with spontaneously in daily life challenges. In management, for instance, decision-making is a crucial task that is essential for achieving organizational goals. From an expert's viewpoint, decision-making incorporates different phases such as defining the problem, identifying possible solutions, selecting criteria, determining the outcomes of each solution, evaluating the solutions, and finally selecting the best solution [97]. When addressing significant decisions, the proper assessment of criteria becomes vital. In such situations, it's essential to base the decision-making process on a systematic approach with explicit evaluation of all involved criteria and leverage appropriate software and tools to compute the solution. In other words, complex decision-making problems are typically addressed with the help of scientific approaches, such as mathematical formulations, statistical methods, and economic theories, typically applied using computer tools that enable the automatic calculation and estimation of solutions [44].

As the importance of decision-making research has grown across various fields, the adoption of Multi-Criteria Decision Making (MCDM) or Multi-Criteria Decision Analysis (MCDA) has become crucial in addressing complex decisions problems [42]. MCDM refers to various techniques used for solving complex decision-making problems. It is an established subject and has been taught for many years at various educational levels and disciplines including industrial engineering, management, and applied mathematics. Moreover, MCDM is well-documented with numerous articles, books, and dissertations, emphasizing its significance and practical applications

[97]. Early concepts on MCDM can be traced back to Benjamin Franklin's "moral algebra" concept [44]. Then, since 1950s, several empirical and theoretical researchers have worked on MCDM methodologies, examining their mathematical modeling capabilities. These studies seek to establish a framework that helps in structuring decision-making challenges and generating preferences from a range of alternatives. Accordingly, several MCDM methods are developed to deal with the structuring, decision-making, and planning phases, especially when several criteria are considered, aiming to find the best solution aligned with the decision-makers' preferences. These methods are categorized into several distinct categories to help understand the available methodologies [44].

Furthermore, MCDM is a well-known subject in the domains of decision science, management, and operations research. In particular, MCDM is one of the fastest-growing areas of operations research and is widely regarded as the most well-known branch of decision-making [42] [98] [99]. It integrates multiple heterogeneous aspects including economic, environmental, political , social, and technical to enable a structured decision-making process. Moreover, MCDM techniques not only find a single perfect solution to decision issues, but also provide a set of feasible solutions, each evaluated against a set of criteria associated with the decision on hand. This provides decision-makers with the ability to select the solution that best matches their specific objectives and the context of the decision. By recognizing the intrinsic complexity of real-world decisions and the varying weight of influencing factors, MCDM serves as a tool for informed and preference-sensitive selection in complex decision scenarios. Likewise, MCDM helps in making informed decisions by providing a systematic process for problem structuring, data collection, criteria definition, and alternative evaluation. Each criterion is weighed to reflect its importance, allowing decision-makers to express their preferences and priorities efficiently. Furthermore, MCDM methods are capable of incorporating both qualitative and quantitative criteria, offering a comprehensive decision-making framework [42] [43]. While criteria based on quantitative variables are typically objective and do not rely on expert opinion, qualitative criteria (variables) are often subject to expert analysis and may be subjectivity. Various approaches such as ranking or scoring systems are used to convert qualitative criteria into quantifiable units that fit within the MCDM framework. Consequently, in the decision-making process, qualitative criteria are quantified through the use of expert-made indicators and measurements [43].

In short, MCDM approaches are used to evaluate and select the most suitable alternatives in the presence of multiple, usually conflicting decision criteria, enabling decision-makers to consider multiple criteria rather than relying on a single criterion [42]. Hence, given a set of alternatives (options) and various decision criteria (conditions or variables), MCDM aims to offer a selection, ranking, description, classification, or sorting, and in most cases, an ordering of choices, ranging from the most preferred to the least preferred alternative [43].

This thesis aims to systematically evaluate job candidates by quantifying their qualifications against specific selection criteria, which include educational background, work experience, and job-related skills. It assigns numerical values to each attribute and criterion, calculates points for every criterion, and then normalizes the points and calculates total score in percentages. These percentages are then translated into a 1–5 star rating scale. The weights of the selection criteria and each candidate's attributes are predetermined. This structured approach ensures a transparent and quantifiable method for assessing candidate suitability.

5.2 Multi-Criteria Decision Making Key Concepts

MCDM includes several elements and concepts based on the nature of the decision-making scenario. Table 6 presents these key concepts, each with a brief description [44] [97].

Table 6 Multi-Criteria Decision Making Key Concepts

Concept	Definition
Alternatives	Represent distinct and usually finite options available to the decision maker to choose from. (Some MCDM problems involve an infinite number of options.)
Criteria	The basis used for evaluating and comparing alternatives to measure their level of success. Criteria can be based on objectives or attributes and it is typically assigned weights of importance.
Objective	The target that is aimed for and worked towards until it is fully realized.
Attribute	The characteristic that an alternative possesses. Each alternative is associated with several characteristics deemed important by the decision maker.
Positive attributes	Desirable attributes where more is better, such as profit or productivity
Negative attributes	Undesirable attributes where less is better, such as costs or losses.
Compensatory attributes	Attributes where the disadvantage of one can be balanced by another.
Non-compensatory attributes	Attributes where the disadvantage of one cannot balanced by another.
Independent attributes	Attributes with no correlation to any other attributes.
Dependent attributes	Attributes with correlation to at least one of the other attributes.
Quantitative attributes	Attributes that can be expressed numerically and are measurable.

Concept	Definition
Qualitative attributes	Attributes that cannot be expressed numerically and are immeasurable.
Aggregation	An alternative's performances on specific criteria.
Decision Variables	Components of an alternative's vector.
Decision Space	Feasible alternatives.
Measures	Elements to quantify an alternative's attribute by assigning numbers or symbols.
Preferences	How an alternative meets a decision maker needs regarding a specific attribute.
Decisions	Variations based on problem type, including choice, ranking, and sorting.

Solving an MCDM problem involves multiple interpretations. The problem can be viewed as selecting the best alternative from a set, or grouping alternatives into various preference sets and choosing a subset. The key concept is to identify nondominated or efficient alternatives that stand out as superior without compromising any of the considered criteria [44].

Mathematically, an MCDM problem can be represented as distinct and finite sets of alternatives (A), evaluation criteria (C) and weights (W) assigned to each criterion based on their significance:

$$A = \{A_i \mid i = 1,2,3, \dots, m\}$$

$$C = \{C_j \mid j = 1,2,3, \dots, n\}$$

$$W = \{w_j \mid j = 1,2,3, \dots, n\}$$

The alternatives (A) are intrinsically homogeneous, but criteria (C) may differ with varying objectives and units. The given mathematical sets represent a simplified form of defining an MCDM problem and they are usually structured as a matrix, as shown in Table 7 [44].

Table 7 Simplified MCDM Matrix

	CRITERIA			
ALTERNATIVES	C_1 W_1	C_2 W_2	… …	C_n W_n
A_1	A_{11}	A_{12}	…	A_{1n}
A_2	A_{21}	A_{22}	…	A_{2n}
…	…	…	A_{ij}	…
Am	A_{m1}	A_{m2}	…	A_{mn}

The matrix's structure might vary depending on the chosen MCDM method. In the given matrix, A_{ij} indicates A_i's value related to C_j. This matrix serves as the basic inputs for MCDM problems [44]. Using an MCDM method, alternatives are then ranked based on their scores from the highest to the lowest [44] [51]. This process encompasses the following general steps [42]:

1. Problem formulation: Define the decision problem, objectives, and constraints.
2. Criteria weighting: Determine the relative importance of each criterion.
3. Alternatives identification: List all potential and feasible alternatives.
4. Evaluation of alternatives: Assess each alternative against the criteria and assign corresponding scores.
5. Aggregation of criteria: Combine the scores for each criterion to formulate an overall score for each alternative using aggregation methods.
6. Sensitivity analysis: Test the impact of varying criteria weights and alternative scores to check the decision-making process's stability.
7. Decision making: Select the alternative with the top aggregated score or a set of top-performing options.

These steps can be summarized as shown in Figure 14.

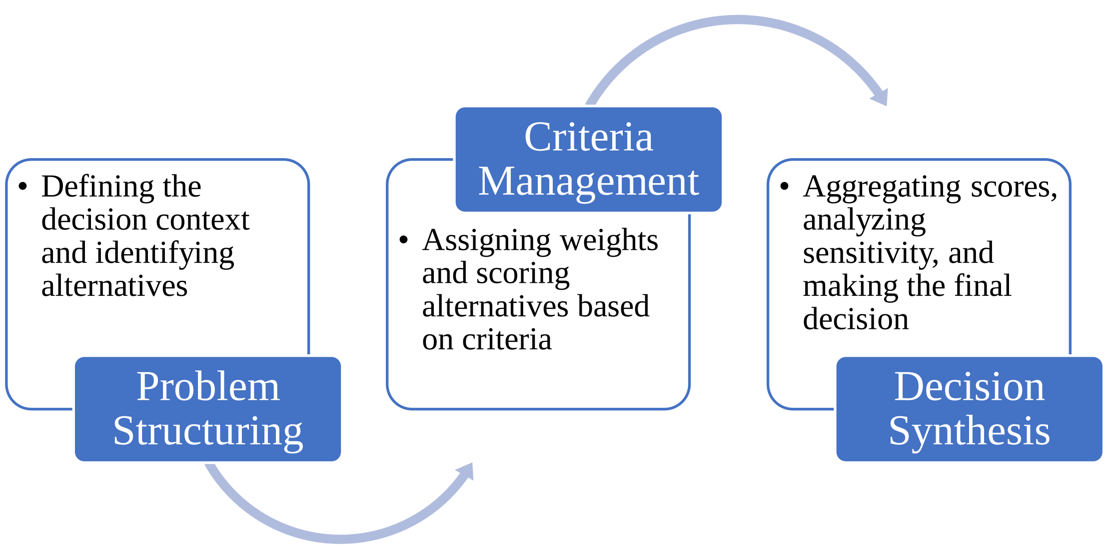

Figure 14 General Steps in MCDM Process

Criteria weights are crucial as they significantly impact the final alternative evaluation. The weights are based on the decision-makers' judgments or derived directly from the decision matrix. Weighting approaches fall into three categories [42]:

1. **Subjective Weighting:** Weights are obtained from decision-makers' opinions. Subjective weighting is usually used with MCDM methods such as Analytic Hierarchy Process (AHP), Analytic Network Process (ANP), Decision Making Trial and Evaluation Laboratory (DEMATEL), Best Worst Method (BWM), Full Consistency Method (FUCOM), and Vital-Immaterial Mediocre Method (VIMM).
2. **Objective Weighting:** Weights are calculated mathematically from the decision matrix. Methods used with this weighting include Entropy, Criteria Importance Through Intercriteria Correlation (CRITIC), Level Based Weight Assessment (LBWA) and Removal Effects of Criteria (MEREC).
3. **Integrated Weighting:** This combine subjective and objective approaches to compute criteria weights and balance both perspectives. Methods examples of this integration include Integrated Determination of Objective Criteria Weights (IDOCRIW) and Objective-Subjective Weighted Method for Minimizing Inconsistency (OSWMI).

5.3 Multi-Criteria Decision-Making Classification

There are several MCDM methods, each come with unique characteristics that are related in many aspects from the type of the problem to the quality of the solution. To choose the right method for specific problems, it's essential to recognize the classification of MCDM problems [44].

MCDM methods are grouped based on different views. One classification is based on the fundamental approach of these methods, such as distance measurement, area comparison and approximation, ratio additive assessment, or the use of algorithms that work under compromising situations. Table 8 presents this broad categorization with respective popular methods and references [51].

Table 8 Popular MCDM approaches and their respective popular methods

MCDM APPROACH	REPRESENTATIVE METHOD	REFERENCE
DISTANCE	TOPSIS	[100], [101]
	EDAS	[102]
AREA	MABAC	[103], [104]
RATIO	ARAS	[105], [106]
	SAW	[107]

MCDM APPROACH	REPRESENTATIVE METHOD	REFERENCE
	COPRAS	[108], [109]
COMPROMISED ALGORITHMS	VIKOR	[110], [111]
	CoCoSo	[112]
	MARCOS	[113]
	RAFSI	[114]

Moreover, MCDM methods are classified in terms of alternative ranking and criteria weighting. Ranking methods have various subcategories, including outranking (e.g., PROMETHEE, ELECTRE), consensus ranking (e.g., VIKOR), combined compromise solution (e.g., CoCoSo), distance-based (e.g., EDAS, MABAC, MARCOS, CODAS, TOPSIS), and pairwise comparison methods (e.g., AHP, ANP), each reflecting advancements in addressing decision-making complexity [42]. Table 9 outlines this classifications of MCDM methods [44].

Table 9 MCDM Classification

Classification Type	Description
Outranking relations	Assesses alternatives by evaluating their outranking degree when one alternative is deemed as good as another.
Utility functions	Considers an alternative's performance across all criteria and helps sort alternatives.
Discriminant function	Linear models with quantitative criteria, not based on preference orders.
Function-free models	Analyzes the overall performance of alternatives using a specific decision rule.
Compensatory/Non-Compensatory	Categories based on whether negative attributes are offset by positive ones.
Individual/Group Decision-Making	Classifies based on the number of decision-makers.
Qualitative/Quantitative Certain/Uncertain	Classifies based on the type of information and whether the information is definite or vague.
Trade off/Non-Trade off	Classifies based on the type of weighting method.
MADM/MODM	Classifies based on the number of alternatives, finite or infinite and whether considering Attributes (MADM) or Objectives (MODM) as criteria type.

In particular, MCDM problems are classified as Multi-Attribute Decision Making (MADM) and Multi-Objective Decision Making (MODM), which is one of the most common categorization [44] [97]. It is based on the type of the problem in terms of finite or infinite number of alternatives [44]. First, MADM focuses on selecting the best alternative from a finite set of alternatives. MADM's problems include elements such as objectives, decision-makers, evaluation attributes, alternatives, variables, and results obtained from comparing alternatives. These elements are structured in the decision matrix, which organizes decision outcomes across alternatives and attributes in a set of rows and columns [97]. In the MADM, objectives, attributes (representing criteria), and alternatives are clear, but the constraints are unclear, and the level of interaction among decision-makers is limited [44]. Second, MODM involves continuous decision-making spaces with an infinite number of alternatives and aims to design the best solution [44] [97]. In the MODM, criteria are objectives, attributes are implicit and while the alternatives are not explicitly defined, the constraints are clear, and there is significant interaction among decision-makers. The MODM represents an optimization problem with no direct and specific alternative chosen as a solution but those lie on a feasible region (decision space) are taken as the solution to the decision-making problem [44]. Since MADM has broader applications than MODM, several MADM methods have been developed by researchers over the past six decades [97]. Figure 15 shows the introduced MCDM methods classification [44].

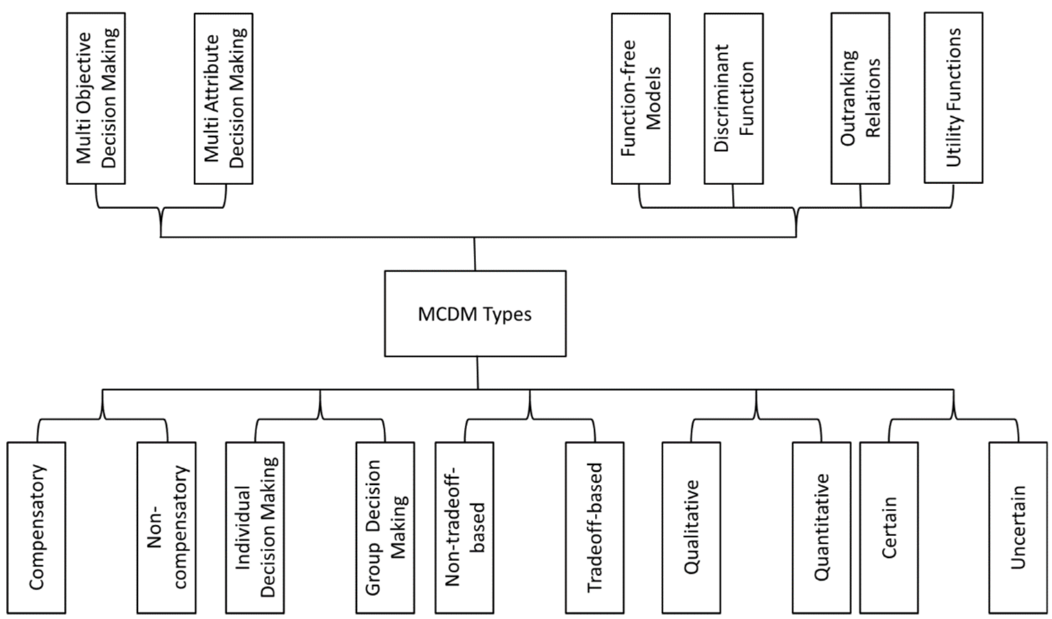

Figure 15 MCDM Classifications

5.4 Multi-Criteria Decision Making Methods

MCDM methods evaluate various qualitative and quantitative criteria that must be addressed to find the best solution. Factors such as cost, price, and process quality frequently feature as the most common criteria in several decision-making scenarios. Furthermore, decision-making problems typically require the involvement of domain experts to assign different weights to the criteria based on their relative importance [44].

Over the past several decades, several methods have been developed or enhanced for solving MCDM problems. The primary distinctions among these methods are related to their computational logic, criteria weighting, applicability, calculation complexities, fuzzy data handling, and data aggregation methodology [44] [51]. Furthermore, each method has its specific advantages and disadvantages. For instance, the AHP method is easy to use but faces challenges arising from the interdependence between criteria and alternatives. Conversely, the Fuzzy Set Theory (FST) method can handle fuzzy inputs, yet it is complicated to use. In generic, all MCDM methods have the advantage of considering inconsistent and conflicting criteria. However, their limitation is that the generated solutions represent a balance between multiple objectives, which may prevent reaching an ideal outcome due to the inherent nature of the issue [44].

Taherdoost and Madanchian [44] listed 60 MCDM methods and identified the 20 most cited ones, with AHP, DEA, FST, TOPSIS, and GP being the main and top cited ones during their study period. They concluded that MCDM methods, particularly MADM methods and notably the AHP method, have become fundamental in decision-making processes that involve multiple criteria. Table 10 lists the 60 MCDM methods with the corresponding number of articles for each, while Table 11 provides descriptions and references for the 20 main MCDM methods [44].

Table 10 MCDM Methods

Method	**Articles**	**Method**	**Articles**	**Method**	**Articles**
Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)	8241	Fuzzy analytic network process (ANP)	586	Complex Proportional Assessment (COPRAS)	445
VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR)	2691	Grey analysis: Grey Relational Analysis/Grey Relational Model (GRA/GRM)	3176	COmbined COmpromise SOlution (CoCoSo)	75
Multi-Objective Optimization by Ratio Analysis (Multi-MOORA)	165	Weighted Sum Model (WSM)	470	Measurement of Alternatives and Ranking according to COmpromise Solution (MARCOS)	35
Multi-Attribute Utility Theory (MAUT)	948	Weighted Product model (WPM)	198	Ranking of Alternatives through Functional mapping of criterion sub-intervals into a Single Interval (RAFSI)	1

Method	Articles	Method	Articles	Method	Articles
AHP	15452	Aggregated Indices Randomization method (AIRM)	4	Automatic Routine Generating and Updating System (ARGUS) method	3
FST	8730	ANP	3126	Lexicographic Method (LM)	311
Case-Based Reasoning (CBR)	3258	Treatment of the Alternatives according To the Importance of Criteria (TACTIC)	1	Measuring Attractiveness by a categorical Based Evaluation Technique (MACBETH)	162
Data Envelopment Analysis (DEA)	9367	Intercriteria Decision Rule Approach (IDRA)	183	Multicriterion Analysis of Preferences by Pair-wise Actions and Criterion Comparisons (MAPPAC)	3
Simple Multi-Attribute Rating Technique (SMART)	646	Evaluation of Mixed Data (EVAMIX)	65	Multi-Attribute Value Theory (MAVT)	315
Goal Programming (GP)	4113	Passive and Active Compensability Multicriteria ANalysis (PACMAN)	3	Best-Worst Method (BWM)	867
ELimination Et Choix Traduisant la REalité (ELimination Et Choice Translating REality) (ELECTRE)	2782	Dominance-based rough set approach (DRSA)	278	Maximax	195
Preference Ranking Organization Method for Enrichment of Evaluations (PROMETHEE)	2715	Characteristic Objects METhod (COMET)	102	An acronym in Portuguese for “Interactive Multi-Criteria Decision Making” (TODIM)	249
Simple Additive Weighting (SAW)	976	Evaluation based on Distance from Average Solution (EDAS)	143	Méthode d’ELimination et de CHoix Includent les relations d’ORdre (MELCHIOR)	0
FUZZY TOPSIS	2014	Multi-Attribute Border Approximation Area Comparison (MABAC)	245	MIN_MAX	22
FUZZY AHP	2804	Additive Ratio Assessment (ARAS)	173	Novel Approach to Imprecise Assessment and Decision Environments (NAIADE)	40
Organisation, Rangement Et Synthese De Donnes Relationelles (ORESTE)	35	REGIonal Multicriteria Elimination (REGIME)	217	Ratio Estimation in Magnitudes or deci-Bells to Rate Alternatives which are Non-Dominated (REMBRANDT)	4
Procédure d’Agrégation Multicritère de type Surclassement de Synthèse pour Evaluations Mixtes (PAMSSEM)	6	TACTIC	10	Multi-Attribute Range Evaluations (MARE)	3

Method	Articles	Method	Articles	Method	Articles
Preference Ranking Global Frequencies in Multicriterion Analysis (PRAGMA)	1267	UTilités Additives (UTA)	31	Weighted Aggregated Sum Product Assessment (WASPAS)	270
QUALIty by FLEXible multicriteria method (QUALIFLEX)	117	Decision making trial and evaluation laboratory (DEMATEL)	1378	DEMATEL-based ANP (DANP)	73
Geometrical Analysis for Interactive Aid (GAIA)	68	Induced Ordered Weighted Averaging (IOWA)	125	KANO model/method (author's name)	476

Table 11 MCDM Main Methods

Method	Description	Original Reference or Underlying Source
AHP	Pairwise comparison of hierarchical criteria considering difference information.	[115]
DEA	Performance assessment of a set of homogeneous DM units with multiple inputs and outputs.	[116]
FST	Quantifying the linguistic facet of accessible data and preferences to address subjective and ambiguous problems.	[117]
TOPSIS	Evaluating based on the distance of alternative to the ideal solution.	[118]
GP	Minimizing the derivation of each objective from the desired target together with optimizing manifold goals.	[119]
CBR	Making recommendations using the analysis of the historical data	[120]
GRA/GRM	Dividing information to white, black, and grey (between known and unknown).	[121]
ANP	A non-linear and more general type of AHP using Markov-chain-based aggregation.	[122]
FUZZY AHP	AHP with the fuzzy evaluation of the alternatives.	[123]
ELECTRE	Outranking the relationship of the alternatives and using pairwise comparison	[124]
PROMETHEE	Outranking method (such as ELECTRE) including several iterations.	[125]
VIKOR	A compensatory version of TOPSIS that is based on minimizing the distance to the ideal solution using a linear normalization approach.	[126]
FUZZY TOPSIS	Based on TOPSIS under a fuzzy environment	[127] [128]
DEMATEL	Verifying relationships/interdependence between variables.	[129]

Method	Description	Original Reference or Underlying Source
PRAGMA	Comparing partial profiles of alternatives considering all the possible criteria pairs.	[130]
SAW	Involving a simple addition of scores representing the goal achievements considering all criteria that is multiplied by the criteria weights.	[131]
MAUT	Based on incorporating uncertainty and risk preferences factors into multi criteria decision support methods.	[132]
BWM	Identifying the best and the worst criteria followed by conducting a pairwise comparisons between each of the best and worst criteria and other ones.	[133]
SMART	weighting the criteria based on their importance and converting importance weights into real numbers.	[134]
Fuzzy ANP	Fuzzy expression of criteria weights in ANP method.	[135]

As introduced, the field of MCDM comprises a wide range of methods, each customized to the complexities of various decision-making problems. A common criticism of MCDM is the realization that applying different MCDM methods to the same problem can produce different results [43] [98] [136]. Moreover, no single method is recognized as the most suitable for every decision-making problem, leading to a scenario where choosing the right MCDM method is an MCDM challenge in itself. Selecting the most appropriate MCDM method is a complex task requiring careful consideration. Several studies have recognized this challenge and conducted various practical comparative analyses of different MCDM methods and other studies have provided guidelines that aid in identifying the most suitable MCDM method for specific decision-making problems [43].

This research focuses on matching and evaluating job candidate qualifications based on established selection criteria encompassing education, experience, and job-related skills. The nature of such decision-making problem is discrete, meaning the candidate qualifications (alternatives or options) are predetermined. This makes MADM methods more applicable to this case. Hence, this thesis concentrates on MADM methods.

In the field of MADM, there are mainly two theoretical perspectives: the American School, which focuses on multi-attribute value functions and Multi-Attribute Utility Theory (MAUT), and the French School, which focuses on outranking methods [43]. MAUT-based methods (such as WSM, WPM, AHP, TOPSIS, COPRAS) typically use a compensatory approach, aiming to combine criteria into a single function that can be maximized, thus facilitating a straightforward determination of the optimal candidate. Conversely, outranking methods such as ELECTRE and PROMETHEE introduce the possibility that some alternatives cannot be directly compared, reflecting a non-compensatory approach. Additionally, it has been observed that ELECTRE and PROMETHEE may not always provide a complete ranking of the alternatives [43]. Given the requirement of this thesis for a full ordering of candidates, ELECTRE and PROMETHEE and similar methods may not be appropriate for addressing the decision-making problem of the thesis and are thus not the focus of this study.

Among MADM methods, this thesis specifically selects the Weighted Sum Model (WSM) its simplicity, relevance, and effectiveness in addressing the complexities of the decision-making scenarios, namely, the selection of the most suitable candidates for specific job positions [3] [45] [46]. The following subsections introduce the concepts of the WSM.

5.5 The Weighted Sum Model

The Weighted Sum Model (WSM), also known as the Simple Additive Weighting (SAW), the weighted linear combination or scoring method is one of the simplest and arguably the most used decision-making method [99] [137] [138]. It is based on Fishburn's weighted average [99]. In WSM, each alternative is evaluated by multiplying its attributes with the corresponding criteria weights. These values are then summed to produce a total score for each alternative. The alternative with the highest score is deemed the best alternative [99]. Essentially, In scenarios with M alternatives and N criteria, the best alternative A* is the one with the highest score obtained from the following expression [137]:

$$A *_{WSM} = max_{M \geq i \geq 1} \sum_{j=1}^{N} a_{ij} \ W_j$$

In this expression, $A *_{WSM}$ represents the WSM score of the best alternative, and a_{ij} denotes the value of the i_{th} alternative with respect to the j_{th} criterion, and W_j indicates the weight of importance of the j_{th} criterion. The WSM is straightforward in situations where all criteria are measured in the same unit, such as profit or time. However, when addressing multidimensional decision-making problems where criteria are measured in different units, using WSM can be conceptually problematic due to its additive utility assumption.

Triantaphyllou and Lin [137]extended WSM model to account for uncertainty inherent in decision-making processes by introducing a fuzzy variant of WSM. In the fuzzy version, both the performance values of the i_{th} alternative and the j_{th} criterion are expressed as fuzzy triangular numbers, denoted respectively as:

$$\widehat{a_{ij}} = (a_{ijl}, a_{ijm}, a_{iju})$$

$$\widehat{w_j} = (w_{jl}, w_{jm}, w_{ju})$$

The values of the criterion weights w_{jm} are structured to collectively sum up to one reflecting the standard practice in non-fuzzy settings where the weights typically add up to one. Consequently, the best alternative according to the fuzzy WSM (FWSM) is determined by the relation:

$$A *_{FWSM} = max \sum_{j=1}^{N} \hat{a}_{ij} \ \hat{w}_j \ for \ i = 1,2,3, \dots, M$$

5.6 Multi-Criteria Decision Making Applications

In the last half-century, the field of MCDM field has witnessed considerable growth, with extensive research contributions. Recently, the scope of MCDM has expanded with the advent of hybrid models that integrate diverse methods for criteria weighting and alternative ranking, thus extending the range of MCDM's applications [42]. This enables the use of MCDM techniques in a variety of decision-making applications, spanning from finance and economics to engineering and healthcare. Table 12 presents some examples of these applications with corresponding references [44] [51].

Table 12 Examples of Various MCDM Applications

Areas of MCDM Methods	Applications Examples	References
Careers and job	Personnel selection, Job choice,	[3], [45], [46], [49], [52], [53], [139]–[142]
Finance and economics	Financial management	[143]–[145]
Waste management	Landfill site analysis, Solid waste management	[146]–[149]
Engineering and production	Material selection, Optimal process parameters	[150]–[153]
Organizations and corporates	Enterprise system selection, Corporate sustainability	[154], [155]
Business process and operations	Business process, Risk and Quality management, Digital marketing	[156]–[159]

Areas of MCDM Methods	Applications Examples	References
Supply chain management	Sustainable supplier selection, green supplier evaluation	[160]–[163]
Energy sector	Renewable energy ranking, Energy policy techniques	[164]–[167]
Civil engineering	Tunnel and Bridge construction analysis, High-speed railway evaluation	[168]–[170]
Building construction and management	Building information modelling, Green technologies selection	[171]–[174]
City and society	Heritage buildings preservation, Smart cities, Water management	[175]–[177]
Education and e-learning	E-learning evaluation, E-learning personalization	[178]–[180]
Transportation	Transportation evaluation design and operation	[181]–[184]
Healthcare	Healthcare technology, quality, and safety assessment	[185]–[188]

Furthermore, Pramanik et al. [51] provided an extensive survey of comparative analysis studies on MCDM methods across various applications. Table 13 lists these studies alongside corresponding applications and references [51].

Table 13 Comparative Analysis of Different MCDM Methods

Ref.	MCDM Methods	Applications Focus	Analysis Performed				
			Sensitivity Analysis	Result Comparison	Statistical Test/Analysis	Rank Reversal	Computation/ Time Complexity
[51]	EDAS, ARAS, MABAC, COPRAS, and MARCOS	Resource selection in mobile crowd computing	√	√	√		√
[136]	ELECTRE, TOPSIS, MEW, SAW, and four versions of AHP	General MCDM problem of ranking	√	√	√	√	
[138]	AHP and SAW	Ranking cloud render farm services	√	√	√		
[189]	TOPSIS, AHP, and COMET	Assessing the severity of chronic liver disease		√	√		
[190]	CODAS, EDAS, WASPAS, and MOORA	Selecting material		√	√		

Ref.	MCDM Methods	Applications Focus	Analysis Performed				
			Sensitivity Analysis	**Result Comparison**	**Statistical Test/Analysis**	**Rank Reversal**	**Computation/ Time Complexity**
		handling equipment					
[191]	TOPSIS, DEMATEL, and MACBETH	ERP package selection	√	√	√		
[192]	AHP, ELECTRE, TOPSIS, and VIKOR	Enhancement of historical buildings		√	√		
[193]	MOORA, TOPSIS, and VIKOR	Material selection of brake booster valve body		√	√		
[194]	AHP, TOPSIS, and VIKOR	Manufacturing process selection		√	√		√
[195]	Multi-MOORA, TOPSIS, and three variants of VIKOR	Randomly generated MCDM problems (i.e., decision matrices) as per [124].	√	√	√		
[43]	WPM, WSM, revised AHP, TOPSIS, and COPRAS	Sustainable housing affordability	√	√	√		
[196]	SAW, TOPSIS, PROMETHEE, and COPRAS	Stock selection using modern portfolio theory		√	√		
[197]	COMET, TOPSIS, and AHP	Assessment of mortality in patients with acute coronary syndrome		√	√		
[98]	SWARA, COPRAS, fuzzy ANP, fuzzy AHP, fuzzy TOPSIS, SAW, and EDAS	Risk assessment in public-private partnership projects	√	√	√		
[198]	WSM, VIKOR, TOPSIS, and ELECTRE	Ranking renewable energy sources	√	√	√		
[199]	WSM, WPM, WASPAS, MOORA, and MULTIMOORA	Industrial robot selection	√	√	√		

Ref.	MCDM Methods	Applications Focus	Analysis Performed				
			Sensitivity Analysis	**Result Comparison**	**Statistical Test/Analysis**	**Rank Reversal**	**Computation/ Time Complexity**
[200]	WSM, WPM, AHP, and TOPSIS	Seismic vulnerability assessment of RC structures	√	√	√		
[201]	AHP, TOPSIS, and PROMETHEE	Determining trustworthiness of cloud service providers	√	√	√		
[202]	TOPSIS and VIKOR	Finding most important product aspects in customer reviews		√	√		
[203]	MABAC and WASPAS	Evaluating the effect of COVID-19 on countries' sustainable development	√	√	√		
[204]	WSM, TOPSIS, PROMETHEE, ELECTRE, and VIKOR	Utilization of renewable energy industry	√	√	√		
[205]	WSM, TOPSIS, and ELECTRE	Flood disaster risk analysis	√	√	√		
[206]	TOPSIS, VIKOR, EDAS, and PROMETHEE-II	Suitable biomass material selection for maximum bio-oil yield		√	√		
[207]	TOPSIS, VIKOR, and COPRAS	COVID-19 regional safety assessment	√	√	√		
[208]	EDAS and TOPSIS	General MCDM problem	√	√	√	√	
[209]	AHP, TOPSIS, ELECTRE III, and PROMETHEE II	Building performance simulation	√	√	√		
[210]	AHP, fuzzy AHP, and ESM	Aircraft type selection		√	√		
[99]	AHP, TOPSIS, and SAW	Intercrop selection in rubber plantations		√	√		

Ref.	MCDM Methods	Applications Focus	Analysis Performed				
			Sensitivity Analysis	**Result Comparison**	**Statistical Test/Analysis**	**Rank Reversal**	**Computation/ Time Complexity**
[45]	AHP, TOPSIS, SAW, and PROMETHEE	Employee placement		√	√		
[211]	TOPSIS, VIKOR, improved ELECTRE, PROMETHEE II, and WPM	Mining method selection		√	√		
[212]	AHP, SMART, and MACBETH	Incentive-based experiment (ranking coffee shops within university campus)		√	√		
[213]	AHP, fuzzy AHP, and fuzzy TOPSIS	Supplier selection		√	√		
[214]	TOPSIS, SAW, VIKOR, and ELECTRE	Evaluating the quality of urban life	√	√	√		√
[215]	AHP, MARE, ELECTRE III	Equipment selection		√	√		
[216]	VIKOR and TOPSIS	Forest fire susceptibility mapping		√	√		
[217]	PIPRECIA, MABAC, CoCoSo, and MARCOS	Measuring the performance of healthcare supply chains	√	√	√	√	
[218]	AHP, AHP TOPSIS, and fuzzy AHP	Mobile-based culinary recommendation system		√	√		√
[219]	TOPSIS, COPRAS, and GRA	Evaluation of teachers		√	√		√
[220]	AHP, TOPSIS, ELECTRE III, and PROMETHEE II	Urban sewer network plan selection		√	√		
[221]	TOPSIS and AHP	Dam site selection using GIS		√	√		

5.7 Multi-Criteria Decision Making Approaches in Personnel Selection

Taherdoost and Madanchian [44] examined the volume of published articles on MCDM methods to assess their importance across various academic fields. Their findings based on searching "ScienceDirect" database for the period from 2012 to 2022 showed a total of 10,116 articles related to "multi-criteria decision-making" keyword and 7,619 linked to "MCDM" keyword. From these findings, 30.60% are associated with MODM and 69.40% with MADM. Moreover, an increase in MCDM publications in recent years was noted, signifying the growing popularity of MCDM methods.

This thesis considers the use of MCDM methods in personnel selection problem.

Hugo Münsterberg (1863 to 1916), an industrial and applied psychologist is often credited as the pioneer in applying ability testing for personnel selection in industrial settings, notably for selecting electric train motormen [3]. With the beginning of World War I, the U.S. Army sought to develop effective methods for selecting the most suitable military personnel using ability tests, leading to advancements in this area during the 1910s, primarily within the military context. Then, a team of psychologists led by Robert Yerkes developed the first group intelligence testing called the Alpha test. This test was used on over 1.7 million recruits during World War I. Accordingly, the field of personnel selection has over one hundred years of history. However, some experts trace the origins of personnel selection through testing back to ancient times, pointing to the Chinese civil servant exams that began in AD 605. These early Chinese exams are believed to be among the first documented instances of personnel selection tests and have influenced subsequent examination systems [3].

Today, the process of selecting personnel is a critical aspect of HRM and crucial for an organization's success. Recognizing that the right personnel are essential for organizational achievement and growth, it's clear that improper personnel can lead to various issues, including resources waste, such as time, effort, and money. Moreover, given the complexity and significance of personnel selection, it's essential to use robust and fair methods in this process [3]. Because personnel selection using conventional "clinical" and testing method is mainly based on the reviews of decision-makers, they therefore subject to biases and may result in wrong selections, thus unsatisfactory hiring. In contrast, leveraging structured and automated methods can enhance efficiency and yield better matches for the final selections. In this context, MCDM approach provides efficient, and reliable methods to address personnel selection challenges. Such MCDM methods are used to facilitate the matching and ranking of candidates, ensuring a more objective and strategic selection process. Moreover, these methods provide valuable support for decision-makers in making their final personnel selection decisions.

The following is a brief review of studies that use MCDM methods in the realm of personnel selection and ranking.

Khorami and Ehsani [3] surveyed the applications of personnel selection by reviewing and categorizing a wide range of research from 1994 to 2014. This comprehensive survey classifies the decision-making approaches used in personnel selection into five categories: Classic MCDM techniques, Fuzzy MCDM techniques, Expert systems models, Grey Relational models, and Hybrid models. Then, it covers numerous methods based on this classification. These approaches are used in various personal selection sectors including healthcare, education, telecommunication, finance, and management. This survey presents an extensive overview of decision-making techniques applied to personnel selection, reflecting the dynamic nature of this research area. Moreover, the study highlights the limitations of traditional techniques such as interviews and traditional tests.

A similar review on the applications of MCDM in personnel selection is conducted by Afshari et al [53]. This comprehensive literature review, covering studies from 1990 to 2010, investigates both classical and fuzzy MCDM methods as well as approaches based on expert systems and Grey system theory. The study presents an analysis of 60 research and summarizes personnel selection approaches and applications. These approaches are applied across various fields, including nurse manager selection, financial analyst selection, and even televised sportscaster selection. The review indicates that the most popular methods in this domain are AHP, TOPSIS, ANP, and expert systems, both in clear and fuzzy environments. Moreover, the paper also explores how the criteria for personnel selection were determined in these studies, noting the use of expert committees and systematic methods such as the Delphi method, Nominal Group Technique (NGT), and brainstorming. However, it also points out that many studies used simplified numerical examples, which may not be applicable in real-life scenarios. One of the critical insights from this review is the identification of limitations in existing approaches. These include the time-consuming nature of AHP for large problems and the extensive questionnaire requirements of the ANP model. Furthermore, most studies neglect possible dependencies between criteria, which could impact the objectivity and quality of decision-making. The review also highlights the little use of Group Decision-Making (GDM) in these studies, despite its importance for comprehensive problem-solving.

Widianta et al. [45] compared TOPSIS, SAW, AHP, and PROMETHEE methods for the application of employee placement based on data of 60 applicants and predetermined criteria of knowledge, skill, ability, physical condition, and attitude. The obtained accuracy levels show a difference due to the varying weighting techniques employed by each method. The varying levels of accuracy of the compared methods are as follows: TOPSIS resulted in 95% accuracy, followed by PROMETHEE at 93.34%, SAW at 81.67%, and AHP at 50%. Despite these differences, all four methods produced similar rankings for the top 10 candidates, with TOPSIS and PROMETHEE aligning closely with expert recommendations, SAW differed in one ranking, and AHP differed in four. AHP's accuracy was found to decrease with a large number of criteria, whereas SAW performed slightly better. The study concluded that TOPSIS is recommended for its superiority in handling multiple criteria, and PROMETHEE stands as an alternative to TOPSIS due to its high accuracy and capability to handle many criteria whereas .

One of the key advantages of the SAW or WSM method in addressing personnel selection challenges is its simplicity and ability to maintain the relative significance of scores [46]. Afshari et al. [46] promoted the use of the SAW method in solving personnel selection problems, emphasizing its effectiveness. They applied the SAW method to a case study in the telecommunications sector and considered seven criteria to identify the best candidate among five potential personnel and ranking them accordingly.

DEMİRCİ [49] used Preference Selection Index (PSI) method for personnel selection to rank a sample of 6 candidates in the business sector based on 7 criteria related to work experience and soft skills. The PSI method ranks alternative based on the concept of overall preference value instead of assigning weights to the criteria. For each alternative, a preference selection index is calculated using statistical concepts and the alternative with the highest preference value is considered the best. When processing a large number of alternatives or criteria, however, the result of PSI can be inconsistent. In this case, criteria weights are used and determined from the decision matrix information.

To evaluate candidates for patient admission roles in private healthcare institutions and offer decision-making guidance, Cakir and Doğaner [52] used the integrated SWARA (Step-Wise Weight Assessment Ratio Analysis) and WASPAS (Weighted Aggregated Sum Product Assessment) methods. Initially, the importance of criteria such as “Computer Software Proficiency,” “Physical Appearance,” and others was assessed by decision-makers from private hospitals and academia, and then the obtained data were analyzed by the SWARA method to determine the weights. The study then applied the WASPAS method at an Istanbul private hospital. Of twelve applicants, five met the requirements and were called for interviews, with recommendations for the best admission personnel provided to the hospital management based on WASPAS analysis.

Li et al. [222] proposed a decision-making framework for personnel selection integrating Data Analytics Algorithms (DAAs) and MCDM methods and considering both subjective judgements of experts and objective patterns found in HR data. The framework includes a data-driven competency-based method, a graph-based ranking aggregation algorithm, and a hybrid MCDM method based on Linear Group Best-Worst Method (LGBWM) and Intuitionistic Fuzzy Numbers (IFNs). The approach is applied to a case study in a Chinese enterprise supported by a decision support system named PLEAS [223]. Moreover, it provides recent summary on personnel selection studies that identifies selection method, weight determination method, and scenarios. Selection methods are divided into decision-making methods (such as TOPSIS, MULTIMOORA, VIKOR, ELECTRE) and aggregation operators methods, which integrate fuzzy information from different experts or criteria. In terms of weight determination, traditional methods such as AHP, ANP, OWA, and SWARA are commonly used together with newer methods such as Best-Worst Method (BWM). The study also provides an analysis on the evaluation criteria used in personnel selection.

Individual job choice is connected to the decision-making problem of personnel selection faced by organizations. While individuals choose roles that align with their personal goals and values, organizations engage in the complex process of identifying and selecting the most suitable candidates for these roles. Selçuk et al. [141] investigate the job choice as a MCDM problem. They use the fuzzy TOPSIS method to analyze the problem. Their study involves a comprehensive survey of 275 participants focusing on seven job choice criteria: wage level, job security, business location, vested benefits, societal respectability of the job, business difficulty level, and the opportunity for flexible working hours. Additionally, four types of employment entities are considered: public corporations, special purpose entities, foreign capital enterprises, and self-employment. The study shows that the most valued criteria among decision-makers (survey participants) are the respectability of the job in society and wage level criteria. Then, it follows the flexibility of working hours, benefits, job difficulty, job location and security. In terms of preferred employment types, foreign capital enterprises come first and it follows their own business, public corporations, and lastly special purpose entities.

A similar study by Pekkaya [142] explores the use of MCDM methods in the career preferences of university students. The study aims to investigate the use of MCDM methods in career preference problem. Pekkaya's study involves a survey administered to university students, focusing on eight professions, and comparatively ranking them using various MCDM methods, including MCGM, TOPSIS, VIKOR, and PROMETHEE. Moreover, the study identifies six critical career selection factors: career opportunities, job security, professional benefits, job flexibility, personal issues, and external influences. Job security emerged as the highest priority, particularly among female students. The study found that MCGM and PROMETHEE results were largely consistent, while VIKOR results varied due to its flexibility in weighting.

Chapter 6

Methodology

6.1 Overview

The Information Systems (IS) is an applied research discipline that applies various theories from other disciplines, such as economics, computer science, and the social sciences, to solve problems intersecting Information Technology (IT) and organizations [224]. In this regard, the main research paradigms used in IS are based on traditional descriptive research emerged from social and natural sciences. Furthermore, interpretive research paradigms are also used in IS to conduct explanatory research. In essence, the research paradigms in IS discipline can be grouped into two main categories: behavioral science and design science [225]. First, the behavioral science research paradigm attempts to develop and verify theories addressing human or organizational behavior. Second, the Design Science (DS) research paradigm is often used in engineering disciplines and computer science as a valid and valuable research paradigm. Since the early 90s, several researchers have used DS paradigm in IS discipline placing explicit value on integrating design as a major component of research. The difference between DS and other research paradigms such as theory building and testing, and interpretive research can be stated as: *"Whereas natural sciences and social sciences try to understand reality, design science attempts to create things that serve human purposes"* [224]. Particular to IT, the DS research paradigm seeks to develop and evaluate IT artifacts that explicitly provide practical solution to solve identified problem extending human and organizational abilities [224] [225]. Several research in IS provided more insights in defining DS research paradigm. These visions include integrating system development into the research process using a methodology that includes theory building, systems development, experimentation, and observations. Other insights define DS as equivalent to traditional social science–based theory building and testing [224].

The previous definition of creating IT artifacts intended to solve an identified problem represents the principle meaning of DS research carried out based on practice rules and guidelines that define the characteristics of well conducted research. This principle follows a rigorous process composed of several activities including designing artifacts, stating contributions, evaluating the design, and communicating the results. It may also involve social, technical, and informational resources innovations. Furthermore, the development of the artifacts should be a search process building on existing theories and knowledge where its value and quality must be rigorously evaluated [224].

To successfully carry out DS research, a methodology is needed to identify the research framework with involved activities. In this context, a methodology is "a system of principles, practices, and procedures applied to a specific branch of knowledge" [224]. It allows to define the framework of the research systematically along with involved steps, objectives, procedures, and outputs. Moreover, it helps in producing and presenting high-quality research.

While the previous paragraph briefly introduced the principles and practices of DS research, procedures represent the third main characteristic of DS research methodology. A procedure should define a generally accepted process for carrying out the research. A process model is introduced in [224] together with principles and practices define a DS Research Methodology (DSRM) serving as a commonly accepted framework for carrying out IS research. The process includes a nominal process (a road map) for the conduct of DS research and a mental model representing characteristics and presentations of research outputs.

Since this research seeks to create software artifacts, it follows the general DSRM introduced in [224]. The introduced DSRM defines a nominal process model for pursuing DS research and a mental model for presenting and evaluating it.

As shown in Figure 16 [224], the DSRM process model incorporates six main activities: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication.

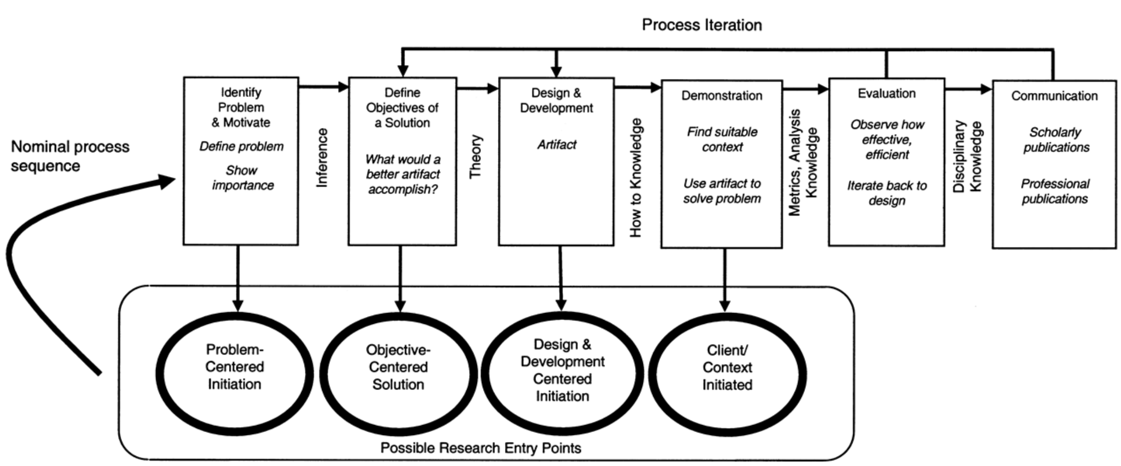

Figure 16 Design Science Research Methodology Process Model

6.2 Design and Development

The overviewed DSRM provides a generic guiding process model where the activities of defining research problem, motivation, objectives, and contribution were introduced in the Introduction chapter. Next, designing and developing a rule engine requires identifying and constructing a KB and an RB to derive new facts. This process requires operating on facts for a certain domain of interest. As the domain of interest, this thesis considers BTM jobs and applicants ontology as the dataset for the job-applicant matching and evaluation application.

In 2009, Information Technology Association of Canada (ITAC) introduced BTM initiative aiming to unify and standardize business and technology educational and professional aspects. Consequently, BTM initiative development involves defining several knowledge areas, learning outcomes, competencies, occupations, career paths and accreditation programs based on several related standards [47] [48].

Based on the BTM occupations and using Protégé ontology editor, Ghebli[46] developed BTM jobs ontology modeling BTM jobs requirements and examples of imported applicants with randomly assigned qualifications. This thesis applies the job-applicant matching and evaluation approach to a modified version of Ghebli's BTM jobs ontology. The modified BTM jobs ontology[47] removes irrelevant concepts that are used in matching in the context of Protégé ontology editor as well as concepts that are not considered in the job-applicant matching and evaluation application developed in this thesis. Accordingly, the research design and development approach include the following main activities:

1. Acquire BTM jobs and applicants RDF ontology modeling jobs instances requirements and applicants instances qualifications with their relations.

2. Analyze BTM ontology to identify possible patterns and criteria for designing the KB and developing the rule engine.

3. Preprocess BTM RDF ontology into the N-Triples (NT) format accepted by Semantic Web Toolkit for Erlang Applications.

[46] https://github.com/JamalElgebli/BTM-jobs-ontology-prototype
[47] https://github.com/MiloudEloumri/match/blob/main/apps/job_match/priv/data/1-btm-jobs-applicants-ontology.rdf

4. Process BTM N-Triples ontology obtained for the previous step using Semantic Web Toolkit for Erlang Applications to eventually convert the ontology into Erlang terms.

5. Based on the obtained Erlang maps, construct jobs facts and applicants facts representing the initial asserted KB as a list of tuples structure accepted by SERESYE and enabling pattern matching the facts based on defined rules.

6. Implement the matching rule engine incorporating the constructed KB and evaluation logic based on WSM with total score and star rating calculation. If a job fact matches an applicant fact, a matching rule is fired and corresponding evaluations are executed, resulting in a new inferred match fact added to the rule engine KB.

7. Based on the structure of the asserted and inferred KB, identify and develop fact templates to quire the KB and obtain corresponding results.

8. Run the application and test querying the rule engine results and state the findings and limitations.

Figure 17 highlights the outlined design and development approach.

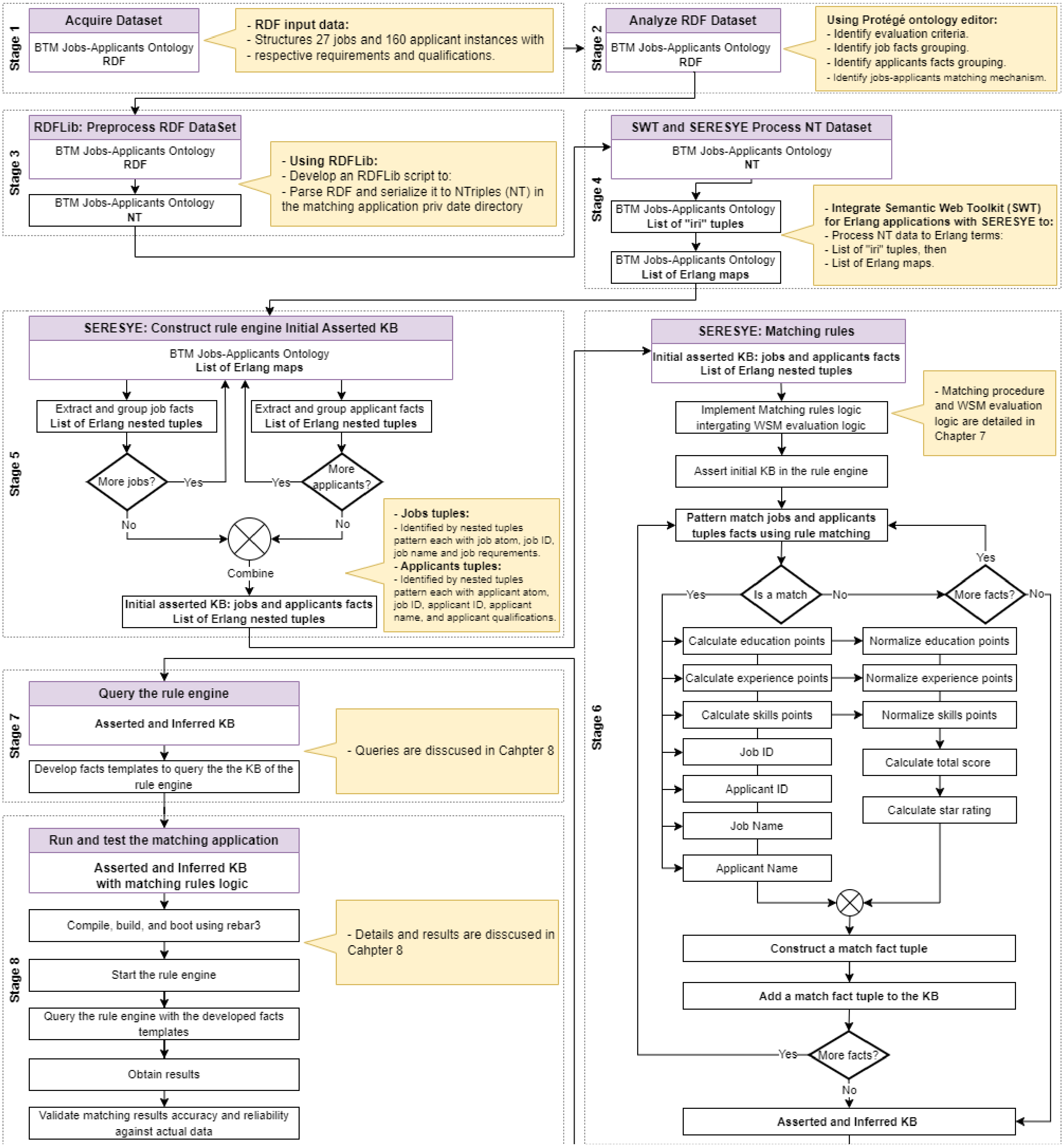

Figure 17 Design and Development Approach

The design and development approach along with examples are detailed in Chapter 7. The following sections overview the outlined design and development approach.

6.2.1 Business Technology Management Jobs Ontology

The used BTM jobs ontology models 27 jobs and 160 applicants individuals with corresponding requirements and qualifications. Figure 18 captured from Protégé ontology editor shows an overview of BTM jobs ontology class hierarchy, and properties, while Figure 19 shows some of the jobs and job seekers individuals.

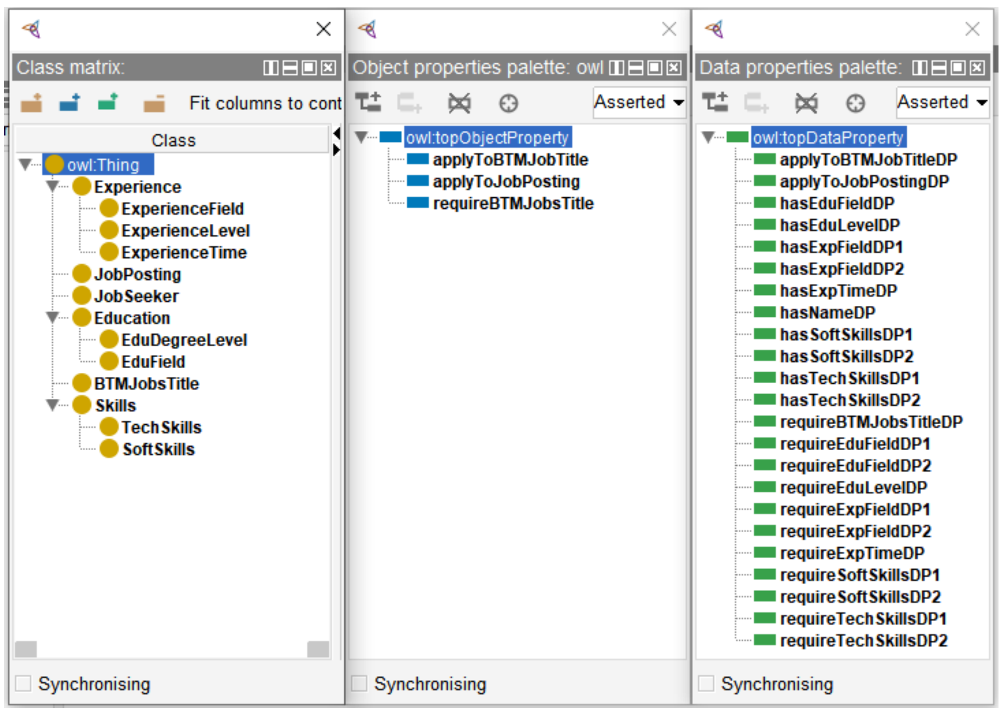

Figure 18 BTM Jobs Ontology Class Hierarchy and Properties

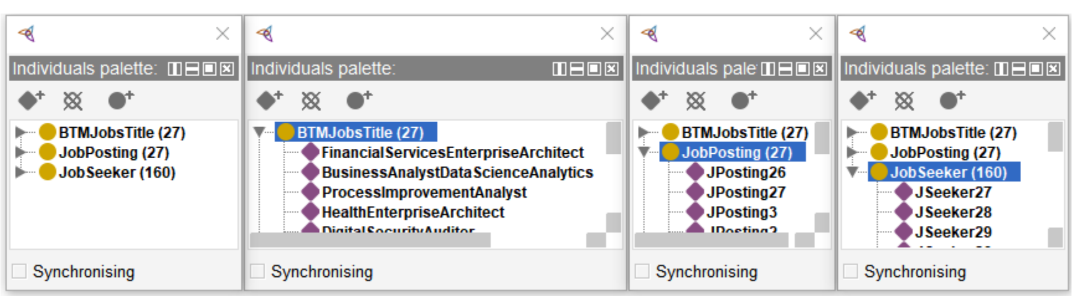

Figure 19 BTM Jobs Ontology Individuals

The following Figure 20 shows top level classes relationships of BTM jobs ontology.

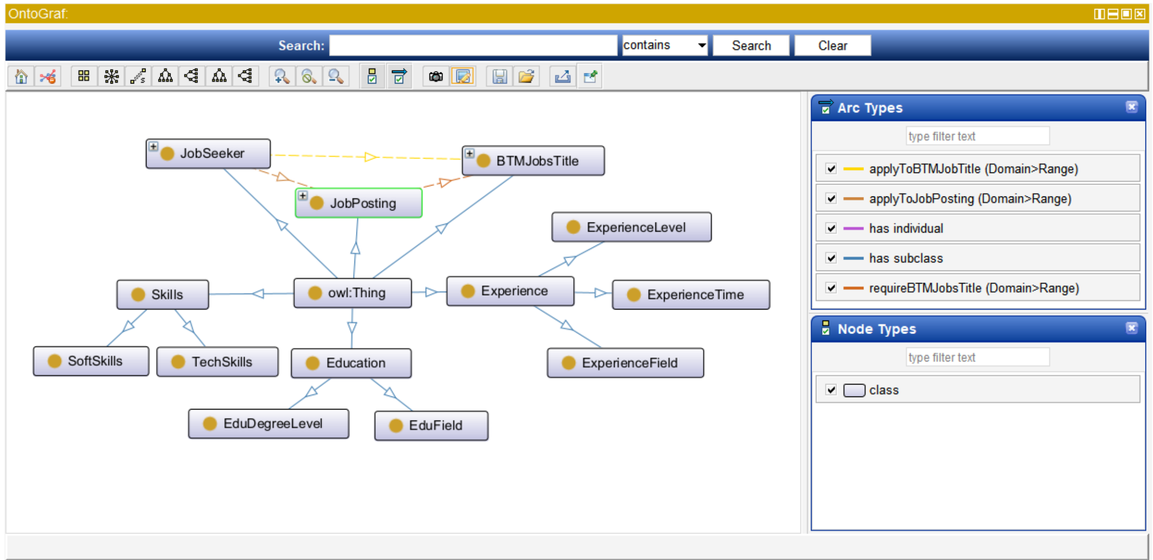

Figure 20 BTM Jobs Classes Relationships

6.2.2 BTM Jobs Ontology Analysis

6.2.2.1 Jobs Data Analysis

In BTM jobs ontology, individuals of "JobPosting" class represent jobs IDs. Each of these individuals require a title representing the name of the job. This relation is captured via the property "requireBTMJobsTitleDP". Figure 21 shows an example of this relation.

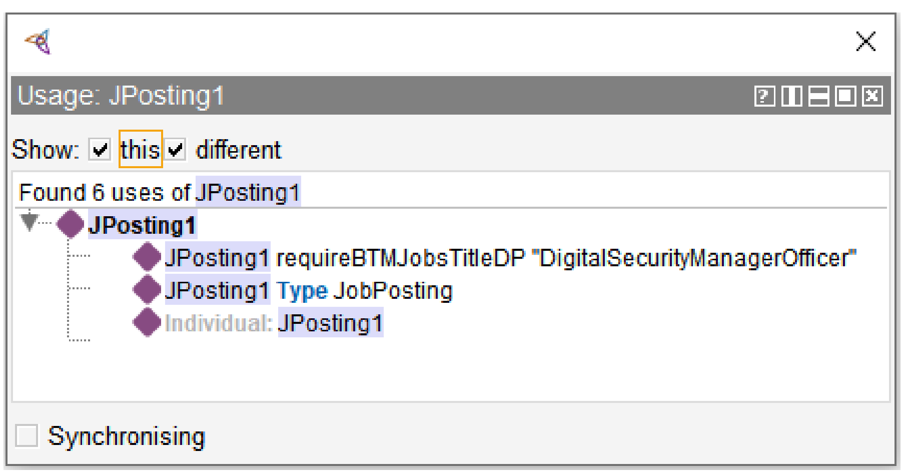

Figure 21 BTM Jobs Ontology requireBTMJobsTitleDP Property Example

This relation specifies that if the Property “P” is “requireBTMJobsTitleDP”, then the Subject “S” holds a job posting ID “JPosting1” and the Object “O” has a job name “DigitalSecurityManagerOfficer”. This pattern is important in identifying and extracting each job posting ID and its corresponding name. The pattern can be simplified in 3-Triple (S, P, O), as shown in the following Figure 22.

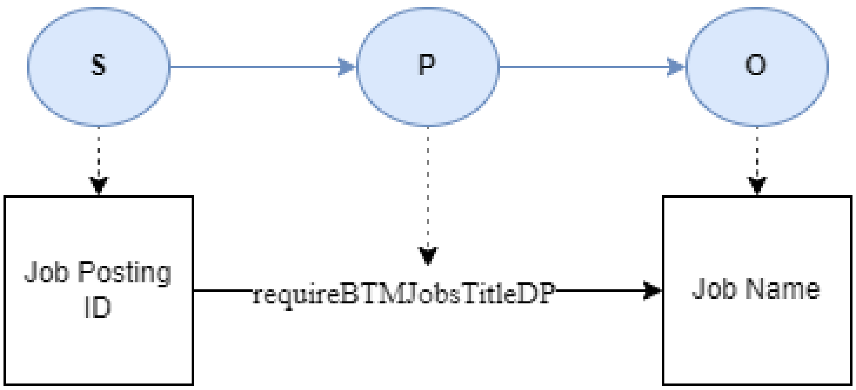

Figure 22 BTM Jobs Ontology Job ID and Name 3-Triple Pattern

The second important jobs pattern is represented by the relation between individuals of “BTMJobsTitle” class and corresponding jobs requirements. Figure 23 shows an example of this relation.

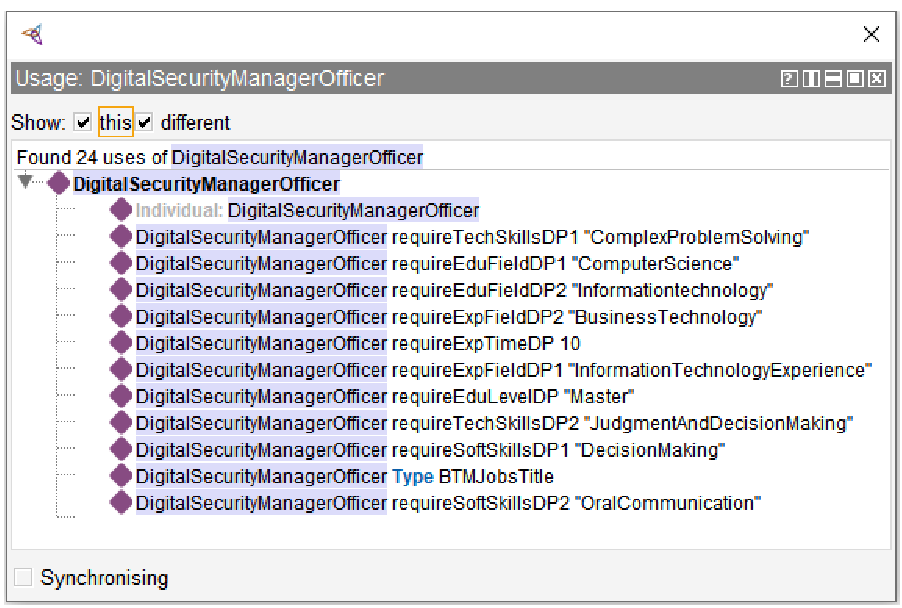

Figure 23 BTM Jobs Ontology Job Requirements Example

As shown on Figure 23, the job titled "DigitalSecurityManagerOfficer" represents the Subject "S" of the relation and the "require" Properties "P" such as "requireEduLevelDP" represent the relation predicates with their corresponding Objects "O" values, such as "Master". This pattern can be depicted in 3-Triple (S, P, O), as shown in Figure 24.

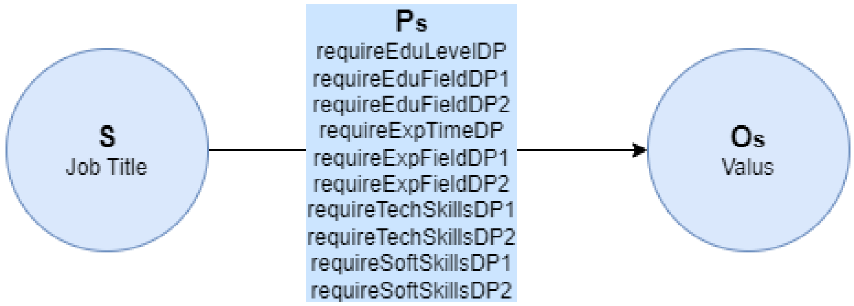

Figure 24 BTM Jobs Ontology Jobs Requirements in 3-Triple Pattern

The relation between a job title and its requirements identifies a pattern that is used to extract and group the requirements for each job based on its title.

6.2.2.2 Applicants Data Analysis

Similar to jobs, individuals of "JobSeeker" class represent applicants IDs. Each of these individuals has a relation with job postings IDs through "applyToJobPostingDP" Property "P". This property allows to link job seekers IDs with their corresponding job posting IDs where job seekers IDs are Subjects "S" and job posting IDs are Objects "O" of the relation, as shown in Figure 25.

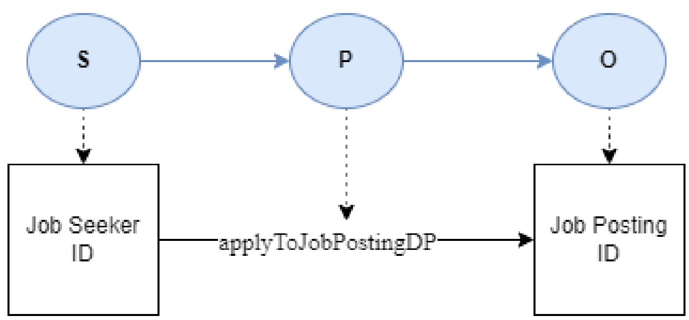

Figure 25 BTM Jobs Ontology Job Seeker ID and Job Posting ID 3-Triple Pattern

Therefore, the "applyToJobPostingDP" property forms an important pattern that is not only used to link job seekers IDs with corresponding job postings IDs, but it is also used in structuring the rule engine facts and rules matching. Such that, a job is matched with a job seeker when their facts share the same job ID.

In addition to the previous important property, each job seeker has a name and qualifications corresponding to jobs requirements except that a job seeker has one educational field instead of two as in the jobs requirements. Figure 26 shows an example of a job seeker qualifications.

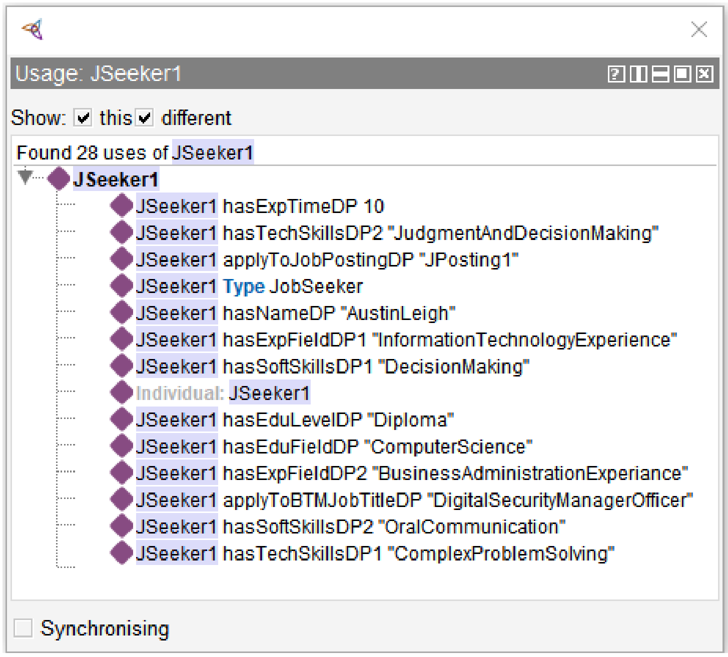

Figure 26 BTM Jobs Ontology Applicant Qualifications Example

In Figure 26, a job seeker or applicant with "JSeeker1" ID represents the Subject "S" of the relations and the "has" Properties "P" such as "hasEduLevelDP" represent the relations Predicates with their corresponding Objects "O" values, such as "Diploma". Moreover, the job seeker properties include "hasNameDP" and "applyToBTMJobTitleDP with their corresponding Objects "O" values: "AustinLeigh" and "DigitalSecurityManagerOfficer" respectively. Figure 27 shows 3-Triple (S, P, O) representation of these relations.

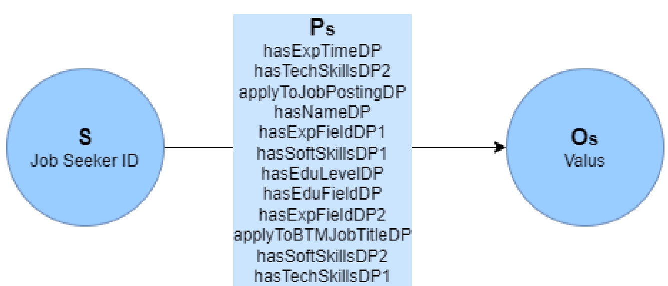

Figure 27 BTM Jobs Ontology Job seekers Qualifications in 3-Triple Pattern

Accordingly, a job seeker qualifications can be identified, extracted, and grouped based on the job seeker ID.

6.2.3 Selection Criteria and Evaluation Method

To evaluate job seekers or applicants qualifications, it requires identifying selection criteria and determining the evaluation method. Based on the analysis of BTM jobs ontology, the selection criteria are represented by Education, Experience and Skills requirements. For education requirements, each job requires two educational fields, and one educational level. For experience requirements, each job requires two experience fields, and one experience time representing years of experience. For skills requirements, each job requires two technical skills and two soft skills. Moreover, each job seeker individual has qualifications corresponding to jobs requirements.

Therefore, these criteria are selected to evaluate job seekers qualifications based on WSM and the matching logic between each job and its corresponding applicants. The WSM is used to evaluate a number of alternatives (job seekers) based on multiple criteria. Each job seeker qualification is assigned points and each criterion is assigned a weight based on its relative importance, and then a score and a star rating are computed for each alternative as the weighted sum of its points on the criteria. The WSM is one of the simplest and most used MCDM method. However, the assignment of points and criteria weights can be subjective and may require HRM experts judgment. This thesis uses a simple approach for assigning points and weights to various criteria. Chapter 7 details the use of WSM and the matching logic.

6.2.4 BTM Jobs Ontology Preprocessing Using RDFLib

To use BTM jobs ontology in the job-applicant matching and evaluation application, it requires preprocessing. This preprocessing is essential to read and save the ontology into a format expected by Semantic Web Toolkit for Erlang Applications. For this purpose, an RDFLib script is developed. RDFLib is a Python library designed to work with RDF formats.

The following outlines a simple use of RDFLib.

1. Installing RDFLib

Before using RDFLib, Python48 and its package manager, pip, need to be installed. Following this, RDFLib49 can be installed using the command: pip install rdflib.

2. Using RDFLib

RDFLib simplifies the parsing and serialization of RDF formats. The following is a simple example50 showing how to read an RDF graph and save it in Turtle format.

```
from rdflib import Graph
# Create a Graph
g = Graph()
# Parse in an RDF file hosted on the Internet
g.parse("http://www.w3.org/People/Berners-Lee/card")
# Save the Graph in the RDF Turtle format in the current directory
g.serialize(destination="tbl.ttl")
```

This script initializes a graph "g". It then parses data from http://www.w3.org/People/Berners-Lee/card and stores it locally in a file named "tbl.ttl" using the Turtle format.

3. Using RDFLib with BTM Jobs Ontology

An RDFLib script[51] is developed to parse the BTM Jobs RDF Ontology and save it in N-Triples format, which is the format accepted by the Semantic Web Toolkit for Erlang Applications. This data preprocessing enables subsequent processing and integration within the job-applicant matching and evaluation application. The developed script is given as follows:

```
# Import necessary libraries

# os module provides a way of using
# operating system dependent functionality
# It is used in this script to:
# construct file paths and
# check if files exist to avoid overwriting
import os

# Imports the rdflib library,
import rdflib

# Imports the Graph class from rdflib for creating RDF graph
from rdflib import Graph

# glob module is used for
```

---

48 https://www.python.org/downloads/windows/
49 https://rdflib.readthedocs.io/en/stable/gettingstarted.html
50 https://rdflib.readthedocs.io/en/stable/intro_to_parsing.html
51 https://github.com/MiloudEloumri/match/blob/main/apps/job_match/priv/data/2-reflib-script.py

```
# retrieving files paths matching a specified pattern
# In this script, it is used to:
# retrieve all files "*" in the input directory
import glob

# Define the directories for input and output
input_dir = "C:\\Users\\milou\\Mirror\\Documents\\me\\4-
Protege\\protege-win\\my-ontologies\\"
output_dir = "C:\\Users\\milou\\match\\apps\\job_match\\priv\\data\\"

# Define Python key-value dictionary specifying
# available output formats and their file extensions
output_formats = {
    "1": {"format": "xml", "extension": ".rdf"},
    "2": {"format": "json-ld", "extension": ".jsonld"},
    "3": {"format": "turtle", "extension": ".ttl"},
    "4": {"format": "n3", "extension": ".n3"},
    "5": {"format": "trig", "extension": ".trig"},
    "6": {"format": "trix", "extension": ".trix"},
    "7": {"format": "N-Triples", "extension": ".nt"},
    "8": {"format": "nquads", "extension": ".nq"}
}

# Prompt user for output format selection
print("Select output format:")
for key, value in output_formats.items():
    print(key + ": " + value["format"]) # Display format options users

# Collect user input for desired output format and overwrite preference
selected_format = input("Enter the number of the desired output format:
")
overwrite = input("Do you want to overwrite existing files? (yes/no):
")

# Validate user input for overwriting files
if overwrite.lower() not in ["yes", "no"]:
    print("Invalid input. Please enter 'yes' or 'no'.")
    exit()

# Process each file in the input directory
for file in glob.glob(input_dir + "*"):
    try:
        g = Graph() # Create a new RDF graph
        g.parse(file) # Parse the current file into the graph

        # Determine the output file format and name
        format_ext = output_formats[selected_format]["extension"]
        format_name = output_formats[selected_format]["format"]

        # Construct the full path and the name of the new output file
        # os.path.basename(file) takes the full path of the current
        # input file being processed, and
        # returns the last part of it, e.g., "file.rdf"
        # split('.')[0] takes the file name from
        # os.path.basename(file), and
        # splits it at every period ("."), and
        # creates a list with each part of the file name
        # is an element in the list.
        # The [0] then selects the first element of this list,
        # which is the name of the file name without its extension.
        # e.g., if the basename of the file is "file.rdf",
        # splitting it on the period gives ["file", "rdf"]
        # Taking the [0] element of this list,
        # results in getting the "file" part
```

```
        output_file = output_dir + format_name + '-' +
os.path.basename(file).split('.')[0] + format_ext

        # Check if overwriting is disabled, and
        # ensure a unique file name
        if overwrite.lower() == "no":
            count = 2 # Initialize a counter to append to the file name
            # os.path.isfile(path) returns True,
            # if the path argument points to
            # an existing file on the file system
            # If the path points to a directory or doesn't exist,
            # it returns False
            # It is used to check if a file in output_file
            # already exists
            # If it exists, the file name is updated to
            # avoid overwriting it
            while os.path.isfile(output_file):
                # Update the file name with an incremented counter
                # to avoid overwriting
                output_file = output_dir + format_name + '-' +
str(count) + '-' + os.path.basename(file).split('.')[0] + format_ext
                count += 1 # Increment the counter

        # Serialize (convert) the graph to the selected format and save
        g.serialize(destination=output_file, format=format_name)

        print(f"The file {file} has been successfully converted to
{format_name} format in {output_file}")

    except Exception as e:
        # Handle any errors during file processing
        print(f"Error occurred while parsing the file {file}:
{str(e)}")
```

This script imports necessary libraries and specifies input and output directories. It then defines a dictionary of output formats, each with a corresponding file extension. Moreover, the script prompts the user to select an output format. It also asks the user whether to overwrite existing files or not. The script then iterates over all files in the specified input directory. For each file, a new RDF graph is initialized and the script attempts to parse the file into the graph. Furthermore, the script constructs the output file name based on the selected format and checks if the file should be overwritten. If the user selects not to overwrite files, the script implements a naming system to avoid duplication by appending a count to the new file's name. The graph is then serialized into the selected format and saved to the output directory. Accordingly, a success message is printed, or an error message if any.

Overall, the script is designed to handle various RDF serialization formats with basic validation for user input as well as error handling via try-except block to report any errors during the parsing and serialization process. The script is run from a command line interface to parse and save the files based on the user selections. The following shows an example of running the script using the command: `python rdflib_script.py`

```
milou@HP122021 MINGW64 ~/Mirror/Documents
$ python rdflib_script.py
Select output format:
1: xml
2: json-ld
3: turtle
4: n3
5: trig
6: trix
7: N-Triples
8: nquads
Enter the number of the desired output format: 7
Do you want to overwrite existing files? (yes/no): no
C:\Users\milou\AppData\Roaming\Python\Python311\site-packages\rdflib\plugins\serializers\nt.py:40: UserWarning: NTSerializer always uses UTF-8 encoding. Given encoding was: None
  warnings.warn(
The file C:\Users\milou\Mirror\Documents\me\4-Protege\protege-win\my-ontologies\1-btm-jobs-applicants-ontology.rdf has been successfully converted to N-Triples format in C:\Users\milou\match\apps\job_match\priv\data\N-Triples-1-btm-jobs-applicants-ontology.nt

milou@HP122021 MINGW64 ~/Mirror/Documents
$
```

The obtained N-Triples ".nt" file is saved in the matching application priv\data directory. This file is then processed in the job-applicant matching and evaluation application by the Semantic Web Toolkit for Erlang Applications resulting in a list of Erlang maps. These maps are then used to construct the KB of the rule engine in the application.

The complete result of preprocessing of BTM jobs ontology from RDF format (1-btm-jobs-applicants-ontology.rdf)) into NT format (3-btm-jobs-applicants-ontology.nt) can be found in the application GitHub repository[52]. Figure 28 shows an example of converting RDF format of "requireBTMJobsTitlDP" relation into corresponding NT format.

[52] https://github.com/MiloudEloumri/match/blob/main/apps/job_match/priv/data/3-btm-jobs-applicants-ontology.nt

```
...
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/ghebl/ontologies/2021/7/Main-btm-ontology#JPosting1">
<rdf:type rdf:resource="http://www.semanticweb.org/ghebl/ontologies/2021/6/btm_positions-ontology#JobPosting"/>
<Main-btm-ontology:requireBTMJobsTitleDP>DigitalSecurityManagerOfficer</Main-btm-ontology:requireBTMJobsTitleDP>
</owl:NamedIndividual>
...
                                                                    RDF

...
<http://www.semanticweb.org/ghebl/ontologies/2021/7/Main-btm-ontology#JPosting1>
<http://www.semanticweb.org/ghebl/ontologies/2021/7/Main-btm-ontology#requireBTMJobsTitleDP>
"DigitalSecurityManagerOfficer" .
...
                                                                    NT
```

Figure 28 RDF Fact and Corresponding NT Fact Example

From Figure 28, it can be noted that NT format simplifies the representation of RDF facts by removing unnecessary structures and restructuring the facts in a simple triple of Subject, Predicate and Object (S,P, O) representation. This NT format is further processed into Erlang terms as explained in the following section.

6.2.5 BTM Jobs Ontology Processing Using Semantic Web Toolkit for Erlang Applications

In this stage, the BTM ontology data in NT format is integrated in the application and processed by the Semantic Web Toolkit for Erlang Applications. The Semantic Web Toolkit for Erlang Applications provides functions to parse this data and represent it in Erlang terms. The process begins by loading the NT file in the matching application. This process involves two subsequent data transformations. First, the NT data is converted into Erlang terms represented by a list of "iri" tuples. This initial transformation captures the raw triples from the NT file. Second, the list of "iri" tuples is converted into a list of Erlang maps. Erlang maps provide more advanced features for data manipulation and iteration. Maps allow for more intuitive access to data elements, support for key-based lookups, and better handling of complex data structures. These features are essential for mapping and extracting jobs and applicants facts and transforming the data into a list of nested tuples accepted by SERESYE and structured in a way that allows to perform pattern matching between facts. Figure 29 illustrates an example of these two data transformations

```
[...
{{iri,<<"http://www.semanticweb.org/ghebl/ontologies/2021/7/Main-btm-ontology#JPosting1">>},
{iri,<<"http://www.semanticweb.org/ghebl/ontologies/2021/7/Main-btm-
ontology#requireBTMJobsTitleDP">>},
<<"DigitalSecurityManagerOfficer">>},
...]
```

Erlang List of Tuples

```
[...
#{c => 1.0, k => <<"AGpmm44UQF8">>,
o => <<"DigitalSecurityManagerOfficer">>,
p => {iri,<<"http://www.semanticweb.org/ghebl/ontologies/2021/7/Main-btm-
ontology#requireBTMJobsTitleDP">>},
s => {iri,<<"http://www.semanticweb.org/ghebl/ontologies/2021/7/Main-btm-
ontology#JPosting1">>},
type => {iri,<<"xsd">>,<<"string">>}},
...]
```

Erlang List of Maps

Figure 29 NT Format Converted into Erlang Terms Example

This data transformation ensures that the BTM job requirements and applicant qualifications are accurately and safely represented in Erlang maps. This step forms the foundation for the subsequent mapping, extraction and representation of jobs and applicants KB that the rule engine can operate on as its asserted facts. This structured and seamless integration with SERESYE enables effective rule processing and decision-making based on the ontology data.

6.2.6 Constructing Rule Engine Knowledge Base

Based on the data obtained in Erlang maps, the next stage is to construct jobs facts and applicants facts, representing the initial asserted KB in a format accepted by SERESYE. This involves creating a list of tuples, where each tuple corresponds to a specific fact about a job or an applicant. Job facts include information such as the atom 'job' to identify jobs tuples , job ID to pattern match with applicants facts, job name, education requirements, experience requirements, and skills requirements. Similarly, applicant facts include details such as the atom "applicant" to identify applicants tuples, job ID to pattern match with jobs facts, applicant ID, applicant name, education qualifications, experience qualifications, and skills qualifications. Jobs facts and applicant facts are extracted and grouped by iterating over the list of maps based on the identified patterns and each is saved separately as a list of tuples with a structure based on the jobs and applicants data analysis that ensures correct mapping between each job and its applicants. Figure 30 shows an example of constructed job and applicant tuple facts structure.

```
[{job,<<"JPosting1">>,
    {<<"DigitalSecurityManagerOfficer">>,
     {{education,<<"Master">>,<<"ComputerScience">>,
           <<"Informationtechnology">>},
      {experience,10,<<"InformationTechnologyExperience">>,
           <<"BusinessTechnology">>},
      {techskills,<<"ComplexProblemSolving">>,
           <<"JudgmentAndDecisionMaking">>},
      {softskills,<<"DecisionMaking">>,<<"OralCommunication">>}}}},
...
]

[{applicant,<<"JPosting1">>,
        {<<"JSeeker1">>,
         {<<"AustinLeigh">>,
          {education,<<"Diploma">>,<<"ComputerScience">>},
          {experience,10,<<"InformationTechnologyExperience">>,
                <<"BusinessAdministrationExperiance">>},
          {techskills,<<"ComplexProblemSolving">>,
                <<"JudgmentAndDecisionMaking">>},
          {softskills,<<"DecisionMaking">>,<<"OralCommunication">>}}}},
...
]
```

Figure 30 Jobs and Applicants Constructed Tuple Facts Structure Example

These two separate lists of tuples are combined in one list representing the initial asserted KB. This combined list of facts ensures consistency and accuracy when matching each job with its applicants. Since there are 160 applicants in the data, 160 corresponding matching rules are expected to be fired resulting in adding 160 inferred match facts to the KB. Each added inferred match is structured in a tuple that identifies the corresponding job and applicant evaluation results.

6.2.7 Implementing the Matching Rules

After constructing the KB, the next stage is to implement the matching rule engine. This engine loads the constructed KB and incorporates evaluation logic based on the WSM. The WSM is used to calculate education, experience, and skills points and percentages as well as the total scores and star ratings for each match between job facts and applicant facts.

The rule engine operates by firing matching rules whenever a job fact matches an applicant fact based on jobs IDs and the constructed tuples structure. When a match is found, the rule engine fires a match rule and performs evaluations according to the WSM. These evaluations result in new inferred facts being added to the KB representing the outcomes of the matching process. Each matching fact is identified by the atom “match”. Figure 31 shows an example of a matching fact.

```
[{match,<<"JPosting1">>,<<"JSeeker1">>,
      {<<"DigitalSecurityManagerOfficer">>,<<"AustinLeigh">>,
       {score,64.81},
       {stars,3.5},
       {education,7,35.71},
       {experience,34.0,58.72},
...
]
```

Figure 31 Inferred Matching Fact Example

The implementation of the matching rule engine[53], including the specifics of the WSM and the evaluation logic, is covered in detail in Chapter 7.

6.2.8 Querying and Testing Rule Engine Results

The final stage involves querying the rule engine to obtain and test the results. This process involves developing fact templates based on the structure of the asserted and inferred KB and calling the query functions provided by SERESYE such as seresye_srv:get_kb/1 and seresye_srv:query_kb/2. For instance, the call to the seresye_srv:get_kb(job_match) function results in returning all facts processed by the rule engine including the asserted and inferred facts with their corresponding matching and evaluation. The call to the function `seresye_srv:query_kb(job_match, {match, <<"JPosting1">>, <<"JSeeker1">>, '_'},` returns corresponding evaluation between a specific job posting "JPosting1" and a job seeker "JSeeker1" if there is match otherwise it returns an empty list "[]".

Such queries with facts templates as arguments are formulated to extract information such as jobs facts, applicants facts, matched job-applicant pairs, with corresponding education, experience and skills evaluations as well as total scores, and star ratings. Therefore, testing the rule engine results involves running various queries to evaluate the accuracy of the matching process. Furthermore, the application includes a module for EUnit tests serving as a template for further testing and validation. Chapter 8 discusses running the application and testing the results.

[53] https://github.com/MiloudEloumri/match/tree/main/apps/job_match

Chapter 7

Job Applicant Matching and Evaluation Foundation

7.1 Overview

Developing a matching model that matches job requirements with applicants qualifications to select the best candidates is a challenging task. It requires understanding job requirements to formulate evaluation criteria and, thus, corresponding matching rules. These rules include the matching logic that enables evaluating applicants according to a weighted level of education and its relevance, a weighted level of experience and its relevance, as well as a weighted number and type of skills in order to match with the corresponding job requirements.

For instance, consider the following simple job rules: If the job requires a bachelor's degree in computer science and the applicant has a bachelor's degree in computer science, then increase their match score. If the job requires 3 years of experience in web development and the applicant has at least 3 years of experience in web development, then increase their match score. If the job requires certain technical skills (e.g., Python, SQL) and the applicant has demonstrated these skills, then increase their match score. If the job requires certain soft skills (e.g., teamwork, communication) and the applicant has demonstrated these skills, then increase their match score. From these examples of rules, it can be noted that their pattern takes the form of simple if-then conditions.

However, the real world of job-applicant matching and evaluation is often more complex. For example, an applicant with fewer years of experience but a higher degree of education might be more suitable for a certain position than an applicant with more experience but less education. Likewise, an applicant with a different but related skill might be able to learn the required skill quickly; thus, the applicant's skill should be positively considered in the matching score. Therefore, the matching model should be adjusted to consider fine-tuned rules according to the specific goals of a certain matching system. This requires designing a coherent matching model operating over multiple requirements and varying qualifications and using weighting strategies, as well as a scoring system to evaluate the match between the job and the applicant. Accordingly, each matching rule contributes a certain number of points to the overall score, and the applicants can be then assessed based on their total score.

The job-applicants matching and evaluation model presented in this thesis aims to assist HRM in selecting the best candidates based on a set of predefined criteria, with a particular focus on education, experience, and skills. Moreover, the model uses WSM concepts and a five-star rating scale, where one star signifies a weak match, whereas five stars indicate an almost perfect match. The star rating is determined by a scoring algorithm that quantifies jobs and the applicants, with each criterion contributing a certain number of points to the overall score. Consequently, the best applicants are evaluated based on their total scores and stars ratings, reflecting the degree of match to the job requirements. The matching model generates a list of matches consisting of applicants points and normalization for each criterion, total score, and stars rating.. Each matched result also includes other information such as corresponding job ID. Moreover, the model describes two matching modes: default mode and strict mode. The default mode is more forgiving, while the strict mode is more restricted in evaluation logic. The concepts of using a five-star rating and two matching modes are inspired by the same concepts used by Canada's Job Match Service.

7.2 Canada's Job Match Service

The Canadian Government has introduced a job match service[54] that helps match foreign workers with job postings from employers across Canada. Operated through the Job Bank of Canada[55], the job match service assigns a star rating from one to five, indicating the degree of matching between the job seeker's profile and the job requirements. A one-star rating signifies a slight match, while a five-star rating indicates a near-perfect match. When a job is posted, the job match service finds job seekers whose profiles meet the job requirements. Employers are then able to view a comparison chart showing how closely the profile meets the job requirements. Employers, then, send job seekers an invitation to interview for the position. Furthermore, the job match service offers three different matching algorithms where users can choose to get matched based on their work history, education, or experience and skills. Furthermore, the job match service[56] provides an option to switch from the default matching mode to the strict matching mode if users are getting too many matches. The strict mode provides results more closely related to the profile details.

[54] https://www.canadim.com/blog/job-match-service-work-canada/
[55] https://www.jobbank.gc.ca/home
[56] https://ns.jobbank.gc.ca/findajob/match

7.3 Matching and Evaluation Model Procedure

Procedure: Match-Evaluate-Jobs-Applicants

Input:

Jobs: A list contains jobs data identified by jobs IDs.

Applicants: A list contains applicants data associated to jobs using jobs IDs and identified by applicant IDs.

Mode: The matching mode: default or strict

Output:

Matched Applicants: A list contains each match between a job and applicant along with the evaluation results including Education Points and Normalization, Experience Points and Normalization, Skills Points and Normalization, Total Score, and Star Rating.

Start:

For each Job in in jobs list

For each Applicant in applicants list

If a Job ID is in an applicant data, a match is found

Evaluate: Education Points and Normalization, Experience Points and Normalization, Skills Points and Normalization, Total Score, and Star Rating.

If Default mode is selected:

Education_Points:

$$Weight[Job\_Required\_Education\_Level] * Applicant\_Education\_Level\_Points + Weight[Job\_Required\_Field\_of\_Study] * Applicant\_Field\_of\_Study\_Points \quad \textbf{(1)}$$

Where: Weight[Job_Required_Education_Level] is weighted as follows:

Diploma:1, Bachelor:2, Master:3 Doctorate:4, Other:0.

Weight[Job_Required_Field_of_Study] is fixed to 2.

Applicant_Field_of_Study_Points is 2 if the applicant field of study exactly matches the job required field of study otherwise it is 1.

Normalized_Education_Points:

$$(Education\_Points - Min\_Points) / (Max\_Points - Min\_Points) * 100 \quad \textbf{(2)}$$

Experience_Points:

$$Weight[Job\_Experience\_Level] * (SUM(Weight[i] * Applicant\_Years\_of\_Experience[i])\ for\ all\ i\ in\ the\ applicant's\ years\ of\ experience) + Weight[Job\_Experience\_Field] * Applicant\_Experience\_Field\_Points \quad \textbf{(3)}$$

Where:Weight[Job_Experience_Level] is weighted as follows:

No Experience: 0, Junior:1 Intermediate:2, Senior:3.If these experience levels are not available in the data, then years range is used as follows: 0 year: 0, 1 year: 1, 2-5 years: 2, higher years: 3.

Weight[i] is 1 for years up to the requirement and 1.5 for years beyond the requirement.

Weight[Job_Experience_Field] is fixed to 2.

Applicant_Experience_Field_Points is 2 if the applicant experience exactly matches the job required experience otherwise it is 1.

Normalized_Experience_Points:

$$(Experience\_Points - Min\_Points) / (Max\_Points - Min\_Points) * 100 \quad \textbf{(4)}$$

Skills_Points:

Assuming each skill is binary (missing: 0 or present: 1),let:

T = {T1, T2, ..., Tm} be the set of required technical skills

S = {S1, S2, ..., Sk} be the set of required soft skills

wT and wS be over all weight for technical skills and soft skills, respectively, where wT + wS = 1 and by default wT = 0.6 and wS = 0.4,

V(Tj) and V(Sk) be the value of the applicant on technical skill Tj and soft skill Sk, respectively

Then, the skills points of the applicant are given by:

$$wT * \Sigma[V(Tj)\ for\ all\ j] + wS * \Sigma[V(Sk) for\ all\ k] \quad \textbf{(5)}$$

Normalized_Skills_Points:

$$Skills\_Points / (wT * m + wS * k)) * 100 \quad \textbf{(6)}$$

where, m and k represent the number of required technical and soft skills respectively.

Total_Score (Arithmetic Mean):

$$(Normalized\_Education\_Points + Normalized\_Experience\_Points + Normalized\_Skills\_Points) / 3 \quad \textbf{(7)}$$

If Strict mode is selected:

Education_Points:

$$(Exact\_Match(Job\_Required\_Education\_Level, Applicant\_Education\_Level) == 1\ ?\ Weight[Job\_Required\_Education\_Level] * Applicant\_Education\_Level\_Points : 0) + (Exact\_Match(Job\_Required\_Field\_of\_Study, Applicant\_Field\_of\_Study) == 1\ ?\ Weight[Job\_Required\_Field\_of\_Study] * Applicant\_Field\_of\_Study\_Points : 0) \quad \textbf{(8)}$$

Where: Exact_Match function is a binary indicator evaluating to 1 or 0 (true or false) and expressed as a ternary operator

Normalized_Education_Points:

$$(Education\_Points - Min\_Points) / (Max\_Points - Min\_Points) * 100 \quad \textbf{(2)}$$

Experience_Points:

$$(Exact\_Match(Job\_Experience\_Years, Applicant\_Experience\_Years) == 1\ ?\ Weight[Job\_Experience\_Level] * (SUM(Weight[i] * Applicant\_Years\_of\_Experience[i])\ for\ all\ i\ in\ the\ applicant's\ years\ of\ experience) : 0) + (Exact\_Match(Job\_Experience\_Field, Applicant\_Experience\_Field) == 1\ ?\ Weight[Job\_Experience\_Field] * Applicant\_Experience\_Field\_Points : 0) \quad \textbf{(9)}$$

Where:Exact_Match function is a binary indicator evaluating to 1 or 0 (true or false) and expressed as a ternary operator

Normalized_Experience_Points:

$$(Experience\_Points - Min\_Points) / (Max\_Points - Min\_Points) * 100 \quad \textbf{(4)}$$

Skills_Points:

$$Skills\_Points = wT * \Pi[V(Tj)\ for\ all\ j] + wS * \Pi[V(Sk)\ for\ all\ k] \quad \textbf{(10)}$$

Where, Π is the product of the sequence.

Normalized_Skills_Points:

$$Skills\_Points / (wT * m + wS * k)) * 100 \quad \textbf{(6)}$$

Total_Score (Harmonic Mean):

$$3 / ((1/Normalized\_Education\_Points) + (1/Normalized\_Experience\_Points) + (1/Normalized\_Skills\_Points)) \quad \textbf{(11)}$$

Star Rating:

0.5 star: 0-10% of the maximum Total_Score

1 star: >10-20% of the maximum Total_Score

1.5 stars: >20-30% of the maximum Total_Score

2 stars: >30-40% of the maximum Total_Score

2.5 stars: >40-50% of the maximum Total_Score

3 stars: >50-60% of the maximum Total_Score

3.5 stars: 60-70% of the maximum Total_Score

4 stars: 70-80% of the maximum Total_Score

4.5 stars: 80-90% of the maximum Total_Score

5 stars: 90-100% of the maximum Total_Score

Matched-Evaluated-Jobs-Applicants

Append a new matching fact and result to the matching list,

Iterate over all data and repeat the evaluation if there is a match between a job and an applicant,

Return Matched-Jobs-Applicants list with evaluation result for each applicant

End.

The above matching procedure can be extended by incorporating related procedures such as scheduling interviews, conducting interviews, extending job offers, and follow up emails.

The following pseudocode provides a simplified representation of the matching and evaluation procedure outlined above:

```
1. INPUT Jobs and Applicants facts (the initial constructed KB)
2. FOR each Job in jobs list DO
3.   FOR each Applicant in applicants list DO
4.     IF Job ID matches Applicant Job ID THEN
5.       IF Default Mode THEN
6.         Calculate Education Points Using Formula (1)
7.         Normalize Education Points Using Formula (2)
8.         Calculate Experience Points Using Formula (3)
9.         Normalize Experience Points Using Formula (4)
10.        Calculate Skills Points Using Formula (5)
11.        Normalize Skills Points Using Formula (6)
12.        Calculate Total Score Using Arithmetic Mean (7)
13.      ELSE IF Strict Mode THEN
14.        Calculate Education Points Using Formula (8)
15.        Normalize Education Points Using Formula (2)
16.        Calculate Experience Points Using Formula (9)
17.        Normalize Experience Points Using Formula (4)
18.        Calculate Skills Points Using Product Formula (10)
19.        Normalize Skills Points Using Formula (6)
20.        Calculate Total Score Using Harmonic Mean (11)
21.      END IF
22.      Calculate Star Rating based on Total Score
23.      Construct Match Fact Tuple
24.      Add Match Fact to the Matched Applicants list
25.    END IF
26.  END FOR
27. END FOR
28. OUTPUT Matched Applicants facts are added to the initial asserted KB.
```

The pseudocode is explained as follows:

Step 1: Input the Jobs and Applicants facts (Asserted KB).

Steps 2-3: Loop through each Job in the jobs list and each Applicant in the applicants list.

Step 4: Check if the Job ID matches the Applicant's Job ID.

Steps 5-12: If in Default Mode, calculate the Education Points, Experience Points, and Skills Points, normalize them, and then calculate the Total Score using respective formulas.

Steps 13-20: If in Strict Mode, perform similar calculations but using strict matching formulas.

Step 22: Calculate the Star Rating based on the Total Score.

Step 23: Construct the Match Fact Tuple.

Step 24: Add the Match Fact to the Matched Applicants list.

Steps 26-27: Continue looping until all Jobs and Applicants facts are processed.

Step 28: Output the Matched Applicants tuples with evaluation results are added to the KB.

The following flowchart depicts the procedure steps.

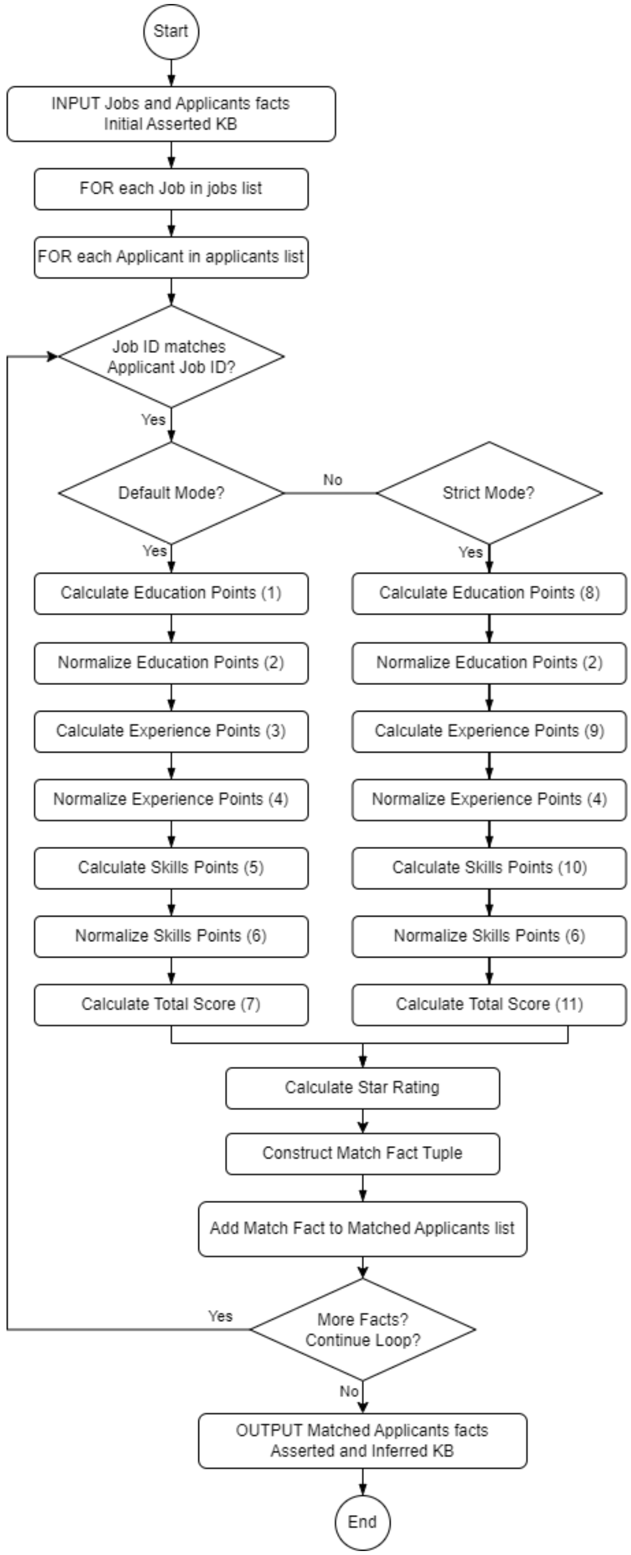

Figure 32 Flowchart of the Match-Evaluate-Jobs-Applicants Procedure

7.3.1 Education Evaluation

Education plays a key role in the overall job-applicant matching and evaluation process. It enables evaluating how closely an applicant's educational qualifications align with the educational requirements of a job based on several criteria. Education criteria cover a wide range of factors, including education level and field.

7.3.1.1 Education Evaluation Criteria

Based on the dataset of BTM jobs ontology and the presented evaluation procedure, education evaluation includes criteria represented by various education levels and fields of study with their corresponding weights. For example, a doctorate degree carries more weight than a master's degree, which in turn carries more weight than a bachelor's degree. Also, an exact match between a required job field of study and an applicant's field of study carries more weight than a mismatch. Moreover, corresponding applicants qualifications are assigned points depending on the relevance of their education at the job required level. Therefore, the first criterion to define is education levels and their corresponding weights and points. This definition may depend on many factors such as experts' judgments or organization's needs. Table 14 shows the education levels weights and points used in this matching model.

Table 14 Education Level Weights and Points

Education Level	*Weight*	*Point*
Diploma	1	1
Bachelor	2	2
Master	3	3
Doctorate	4	4
Other	0	0

The defined education weights and applicant points assume that higher levels of education are more valuable. Thus, it gives more points to applicants who greatly exceed the job-required education level, which is not always ideal. The weights, however, can be adjusted to meet specific requirements. For instance, the weights can be fed as parameters to the matching model education input function, allowing system users to input the weights based on their specific needs. Another option is to only assign points to applicants if their education degree exactly matches the job-required degree, which is the case in the Strict mode of the presented matching model.

The second criterion deals with the weights and points of fields of study. Similar to education levels, the weights and points assigned to fields of study depend on several alike factors. In this matching model, the weight of the required field of study is fixed at 2. For the applicant field of study points, it is challenging to consider several possible related fields of study. For simplicity, the applicant's field of study is assigned a value of 2 points if it exactly matches the required field of study and a value of 1 point otherwise, as shown in Table 15.

Table 15 Education Field Weights and Points

Education Field	*Weight*	*Point*	*Value*
Exact Match	2	2	4
Other	2	1	2

This simplification facilitates the matching process while still accounting for the relevance of the applicant's field of study. Moreover, while the assigned values are used as defaults, they can be changed to meet specific requirements.

Accordingly, the applicant's education points are evaluated as the weighted sum of their points on the education level and field of study, as shown in Formula (1) and Formula (8) for the Default and Strict modes, respectively:

$$Default\_Education\_Points =$$

$$Weight[Job\_Required\_Education\_Level] * Applicant\_Education\_Level\_Points + Weight[Job\_Required\_Field\_of\_Study] * Applicant\_Field\_of\_Study\_Points \quad (1)$$

$$Strict\_Education\_Points =$$

$$(Exact\_Match(Job\_Required\_Education\_Level, Applicant\_Education\_Level\_Points) == 1\ ?\ Weight[Job\_Required\_Education\_Level] * Applicant\_Education\_Level\_Points : 0) + (Exact\_Match(Job\_Required\_Field\_of\_Study, Applicant\_Field\_of\_Study) == 1\ ?\ Weight[Job\_Required\_Field\_of\_Study] * Applicant\_Field\_of\_Study\_Points : 0) \quad (8)$$

To express the education points in a more meaningful percentage way, the raw points are normalized as shown in Formula (2):

$$Normalized\_Education\_Points =$$

$$(Education\_Points - Min\_Points) / (Max\_Points - Min\_Points) * 100 \quad (2)$$

7.3.1.2 Education Evaluation Example

To illustrate the education evaluation, consider the job requirements and applicant qualifications presented in Table 16.

Table 16 Job-Applicant Example

	Education		**Experience**		**Skills**	
	Level	Field	Years	Field	Technical	Soft
Job	Bachelor	Computer Science	2	Web Development	Python, SQL	Teamwork, Communication
Applicant	Master	Computer Science	3	Web Development	Python	Teamwork, Communication

Evaluation of Education Points in Default Mode:

From Table 16, the job requires a bachelor's degree in computer science, and the applicant has a master's degree in computer science. By applying the respective weights and points defined in Table 14, the first education criterion evaluates to 2 * 3 = 6. Moreover, based on the weights and points defined in Table 15, the second education criterion evaluates to 2 * 2 = 4 since the job requires a field of study in computer science, and the applicant's field of study matches this field. From formula (1), therefore, the total education points evaluate to:

$$Default\_Education\_Points = 2 * 3 + 2 * 2 = 10\, points$$

Normalization of Education Points:

To express the education points as a percentage, the raw points need to be normalized. This normalization requires determining the minimum and maximum possible points, as shown in Formula (2). First, the minimum points (Min_Points) correspond to 2 points. This value is obtained by considering that any applicant will receive 0 points for missing a degree and a minimum of 2 points when their field of study does not exactly match the job requirement. Thus, from Formula (1), the minimum possible points are: 2 * 0 + 2 * 1 = 2 points. Second, the maximum points (Max_Points) are determined based on specific job requirements. In the given example, the maximum points equal to 12 points. This value is obtained as follows: (1) the job requires a bachelor's degree, which has a weight of 2; (2) the highest possible degree is a doctorate, which is assigned 4 points; (3) the weight of the required field of study is fixed at 2; (4) the applicant field of study points are 2 points at their maximum; hence, for Formula (1), the maximum possible points are: 2 * 4 + 2 * 2 = 12 points for this particular job-applicant example. After obtaining the raw education points as well as the minimum and maximum points, the normalized education points can be calculated based on Formula (2) as follows:

$$Normalized\_Default\_Education\_Points\ =\ (10 - 2)\ /\ (12\ -\ 2)\ *\ 100\ =\ 80\%$$

It can be noted that the Min_Points will always evaluate to 2 points regardless of any job requirements or applicant qualifications, whereas the Max_Points will vary depending on a specific job-required degree. Because Min_Points always evaluate to 2 points, the normalized education points can be expressed as: $(Eduction\_Points\ -\ 2)\ /\ (Max\_Points\ -\ 2)\ *100$. However, the use of Min_Points enhances clarity and accounts for any possible future changes in how to obtain the Min_Points, which represent a variable instead of a constant 2.

Evaluation of Education Points in Stric Mode:

The education points in the Strict mode are evaluated using Formula (8), which is essentially the same as the formula used in the Default mode, with the addition of points exclusion based on exact matches between the job requirements and the applicant's qualifications. This exclusion is imposed by the Exact_Match function, which evaluates to 1 (true) or 0 (false) based on the presence or absence of an exact match. To explain, The Exact_Match function in Formula (8) is expressed as a ternary operator. It checks whether or not the job requirements are exactly matched by the applicant's qualifications . If there is an exact match, the function returns 1, otherwise it returns 0. Then, the function returned value is multiplied by the weighted points calculation for each category (education level and field of study). Hence, if Exact_Match returns 1 (true), then calculate the weighted points the same as in the Default mode. If Exact_Match returns 0 (false), then the points of the corresponding category are zeroed out. Therefore, any mismatch, either in educational level or field of study, results in zeroing its points. The use of the ternary operator is common in several programming languages, and it can be simplified as: $(condition)\ ?\ a\ :\ b$. It simply specifies that if the condition evaluates to true, return the first result after the "?" sign, which is "a" or else return the second result after the ":" sign, which is "b". The Exact_Match function is expressed as a ternary operator to signify and clarify its return value (1 or 0); otherwise, the function can be simplified as follows:

$$Exact\_Match(Job\_Required\_Education\_Level, Applicant\_Education\_Level)$$
$$* \; Weight[Job\_Required\_Education\_Level] \; * \; Applicant\_Education\_Points$$
$$+ \; Exact\_Match(Job\_Required\_Field\_of\_Study, Applicant\_Field\_of\_Study)$$
$$* \; Weight[Job\_Required\_Field\_of\_Study] \; * \; Applicant\_Field\_of\_Study\_Points$$

Following the job-applicant example presented in Table 16, the Strict mode based on the return value of the Exact_Match function and assigned weights as well as points results in the following strict education points:

$$Strict\_Education\_Points \; = \; (0 \; * \; 2 \; * \; 3) \; + \; (1 \; * \; 2 \; * \; 2) \; = \; 4 \, points$$

It can be noted that the education points for evaluating the education level criterion are zeroed out as a result of the 0 returned by the Exact_Match function indicating a mismatch between the job-required degree (bachelor) and the applicant's degree (master), whereas the second call to the Exact_Match function for evaluating the education field criterion returns 1 signifying the exact match between the job-required field (computer science) and the applicant's field (computer science). Applying normalization to the strict education points is the same as applying it in the default mode, and it results in the following percentage:

$$Normalized\_Strict\_Education\_Points \; = \; (4 \; - \; 2) \, / \, (12 \; - \; 2) \; * \; 100 \; = \; 20.00\%$$

From the example, it can be seen that there is a significant difference between the Default mode and Strict mode results. The Default mode results in 10 points, corresponding to an 80.00% score, while the Strict mode results in 4 points, corresponding to a 20.00% score.

7.3.1.3 Education Criteria and Evaluation Methods

The introduced education MCDM-WSM evaluates applicants for jobs based on how closely their educational level and field align with the corresponding educational requirements of the job. The model is based on the concept of applying weights and points to these two criteria as well as using Default and Strict modes, all of which allow for more fine-tuning of evaluation results. This section summarizes the introduced education evaluation approach and also highlights other related criteria and evaluation methods.

The Significance of Weighting in Education Criteria:

In the presented matching model, different weights and points are assigned to education level and fields of study to reflect their differences. These weights and points help to distinguish the relative value of educational qualifications. Additionally, users of the matching model can specify and fine-tune these weights and points depending on their specific criteria.

Default and Strict Modes of Education Evaluation:

The introduction of Default and Strict modes in the matching model allows for obtaining loose or tight matching results. In the Default mode, the matching model increases education level points proportional to the job-required level and the applicant's level, unless the applicant has no degree. It also doubles field of study points if it is an exact match. Conversely, in the Strict mode, the matching model zeros out education level and/or field of study points if either is not an exact match.

Additional Criteria in Education Matching:

While the presented matching model focuses on the education level and fields of study criteria, other education criteria can be considered, including the following:

1. **Certificates, Licenses or Training**: in some fields, relevant certifications or licenses can be very important, such as certifications in IT, or licenses to practice law or medicine, etc.
2. **Institutions**: some employers may prefer applicants from certain highly ranked institutions or those accredited by specific bodies.
3. **Grades or GPA**: some employers may consider the applicant's academic performance such as grades, or GPA, though this is usually less relevant for experienced workers.

These criteria and others are not considered in this matching model. Moreover, in the introduced matching model, education level criterion is based on the applicant highest level of education, yet all applicant levels of education can be considered as follows:

$$Education\_Points = SUM(Weight[i] * Applicant\_Education\_Points[i]) \quad for\ all\ i\ in\ the\ applicant's\ education\ levels$$

Alternative Education Points Evaluation Methods:

Apart from the presented WSM evaluation approach, alternative simple methods for evaluating education points can be considered such as simple comparison and percentage-based comparison. In simple comparison, each education criterion is assigned certain points. The evaluation is then the difference between the applicant's education points and the job's required education points. For instance, if a job requires a bachelor’s degree (assigned 2 points) and an applicant has a master's degree (assigned 3 points), then education level points are: $3 - 2 = 1$. In percentage-based comparison the same points assignment is used but the evaluation is based on the ratio of the applicant's education points to the job's required education points. The percentage-based comparison method may result in a score over 100% if the applicant's education exceeds the job requirements. For example, following the previous example the education level points are: $(3/2) * 100 = 150\%$. As presented in the MCDM chapter, several other evaluation methods can be used.

Education's Role in Job-Applicant Matching:

While education is an important criteria in the job-applicant matching process as it can be a good indicator of an applicant's background and potential, it does not necessarily reflect their practical experience and skills as well as their ability to perform the job. Therefore, criteria of experience and skills are equally important in the overall job-applicant matching and evaluation process.

7.3.2 Experience Evaluation

Experience plays a key role in the process of evaluating the match between job requirements and applicants qualifications. Moreover, experience is a broad term that encompasses not only actual work history but also professional development and achievement. For instance, it may include projects completed, responsibilities held, tools and methodologies learned, problems solved, promotions earned, etc. Furthermore, there are several common experience or position levels that are generally understood in the job market, such as Junior, Intermediate, Senior, etc. The actual definitions and responsibilities of these positions can vary by industry, company, and role. Likewise, determining the minimum and maximum years of experience associated with each position level depends on various factors.

7.3.2.1 Experience Evaluation Criteria

According to BTM jobs ontology, experience evaluation is based on experience years and field criteria. These criteria are assigned weights and points reflecting their significance based on specific needs. The value of an applicant's experience on a particular criterion is determined by the levels or years and relevance of their experience. Table 17 shows the experience levels, weights, and years ranges considered in the presented matching model.

Table 17 Experience Levels, Weights, and Ranges

Experience Level	*Weight*	*Years Range*
No Experience	0	0
Junior	1	1
Intermediate	2	2-5
Senior	3	6-15

Moreover, each year of experience is given 1 point for years up to the requirement and 1.5 points for years beyond the requirement.

For field of experience criteria, assigning specific weights and values to various related experience fields is challenging. To simplify the process, the matching model adopts the approach of assigning a fixed weight of 2 to the job-required experience field as well as 2 points to the applicant's experience field if it exactly matches the job experience field and 1 point otherwise, as shown in Table 18.

Table 18 Experience Field Weights and Points

Experience Field	*Weight*	*Point*	*Value*
Exact Match	2	2	4
Other	2	1	2

Accordingly, the experience points of an applicant are calculated as the weighted sum of their experience level/years and field values, as shown in Formula (3) and Formula (9) for the Default and Strict modes, respectively:

$$Default\_Experience\_Points =$$

$$Weight[Job\_Experience\_Level] * (SUM(Weight[i] * Applicant\_Years\_of\_Experience[i])\ for\ all\ i\ in\ the\ applicant's\ years\ of\ experience) + Weight[Job\_Experience\_Field] * Applicant\_Experience\_Field\_Points \quad \textbf{(3)}$$

$$Strict\_Experience\_Points =$$

$$(Exact\_Match(Job\_Experience\_Years, Applicant\_Experience\_Years) == 1\ ?\ Weight[Job\_Experience\_Level] * (SUM(Weight[i] * Applicant\_Years\_of\_Experience[i])\ for\ all\ i\ in\ the\ applicant's\ years\ of\ experience) : 0) + (Exact\_Match(Job\_Experience\_Field, Applicant\_Experience\_Field) == 1\ ?\ Weight[Job\_Experience\_Field] * Applicant\_Experience\_Field\_Points : 0) \quad \textbf{(9)}$$

To adjust the raw experience points in terms of percentage, the points are normalized as shown in Formula (4).

$$Normalized\_Experience\_Points =$$

$$(Experience\_Points - Min\_Points) / (Max\_Points - Min\_Points) * 100 \quad (4)$$

7.3.2.2 Experience Evaluation Example

The job-applicant example given in Table 16 requires 2 years of experience in Web Development, and the applicant has 3 years of experience in Web Development. The experience points are calculated in the Default and Strict modes as follows:

Evaluation of Experience Points in Default Mode:

Based on the introduced matching procedure, the experience points are evaluated as follows. Each year of experience up to the requirement is assigned 1 point, and each year beyond the requirement is assigned 1.5 points. Moreover, the weight of 2 years of experience is 2, the weight of the required experience field is 2, and the applicant's experience field value is 2 points if it matches the job's required experience field; otherwise, it is 1 point. Therefore, experience points in Formula (3) evaluates to:

$$Default\_Experience\_Points = 2 * (2 * 1 + 1 * 1.5) + 2 * 2 = 11\, points$$

Normalization of Experience Points:

Normalizing the obtained 11 experience points requires determining the possible minimum and maximum points. Based on the presented experience criteria weights and points and from Formula (3), the possible minimum points are 2 points obtained from (0 + 2 * 1 = 2) for applicants with 0 years of experience and mismatched experience field. Furthermore, based on the previous discussion on the levels and the range of years of experience, the possible maximum points are determined as follows: (1) The job requires 2 years of experience with a weight of 2; (2) The highest years of experience is 15; (3) the job-required 2 years of experience are worth 1 point each, and thus the remaining possible 13 years are worth 1.5 points each; (4) The job experience filed weight is fixed to 2, and the applicant's filed study points are 2 at their maximum. Therefore, Formula (3) results in the following maximum experience points for this particular job-applicant example: 2 * (2 * 1 + 13 * 1.5) + 2 * 2 = 47.

The normalization of the obtained 7.5 experience points, hence, evaluates to:

$$Normalized\_Default\_Experience\_Points = (11 - 2) / (47 - 2) * 100 = 20.00\%$$

Evaluation of Experience Points in Strict Mode:

The Exact_Match function shown in Formula (9) for strict experience points works in the same way as the Exact_Match function discussed in the education matching section. Given the job-applicant example, Formula (9) and Formula (4) result in the following strict experience points and normalization respectively:

$$Strict\_Experience\_Points = 0 + 2 * 2 = 4\,points$$

$$Normalized\_Strict\_Experience\_Points = (4 - 2) / (47 - 2) * 100 = 4.44\%$$

It can be noted that the experience points for the experience years criterion are zeroed out as a result of the mismatch between the job-required experience (2 years) and the applicant's experience (3 years). Moreover, the Default mode results in 11 points representing a 20.00% score, whereas the Strict mode results in 4 points representing a 4.44% score.

7.3.3 Skills Evaluation

Skills play a crucial role in evaluating an applicant's match for a specific job. Skills are typically categorized into two main types: technical skills and soft skills. Technical skills incorporate the specific knowledge, abilities, and expertise required to perform tasks related to a particular job. Soft skills, on the other hand, refer to the interpersonal and personal attributes that contribute to effective communication, teamwork, and other non-technical qualities.

7.3.3.1 Skills Evaluation Criteria

Based on these two categories, the skill points in the presented matching model are evaluated similarly to education and experience points, but with a focus on matching the specific skills required by the job, where each skill is a criterion. This is achieved first by using binary values (0 or 1) for each skill. For example, if an applicant demonstrates a required skill, 1 point is assigned; otherwise, 0 is given. Second, to introduce some variation based on technical skills and soft skills categories as well as the importance of each category to certain jobs, an overall weight is used for each category, where technical skills are assigned a weight of 0.6, and soft skills a weight of 0.4 as shown in Table 19.

Table 19 Skills Weights and Points

Skills	*Weight*	*Point*
Technical Skills	0.6	1/0 for each skill
Soft Skills	0.4	1/0 for each skill

As shown in the matching procedure, skills are evaluated based on Formula (5) and Formula (10) for the Default and Strict mode respectively:

$$Default\_Skills\_Points = wT * \Sigma[V(Tj)\ for\ all\ j] + wS * \Sigma[V(Sk) for\ all\ k] \quad (5)$$

$$Strict\_Skills\_Points = wT * \Pi[V(Tj)\ for\ all\ j] + wS * \Pi[V(Sk)\ for\ all\ k] \quad (10)$$

The normalization of skills points is given in Formula (6):

$$Normalized\_Skills\_Points = Skills\_Points / (wT * m + wS * k)) * 100 \quad (6)$$

7.3.3.2 Skill Evaluation Example

The job example presented in Table 16 requires technical skills of Python and SQL as well as soft skills of Teamwork and Communication while the applicant has all the skills except SQL technical skill, therefore, the skills evaluation in default and strict mode are given as follows:

Evaluation of Skills in Default Mode:

In the Default mode, skills points are assigned based on the presence or absence of each required skill. Each skill criterion is given a binary value of 1 if the applicant possesses the skill and 0 otherwise. For technical skills, the applicant has the required Python skill, thus the value for Python is 1, while the value for SQL technical skill is 0 as the applicant does not demonstrate it. Similarly for soft skills, the applicant possesses the two required soft skills, Teamwork and Communication, therefore each skill has a value of 1. Then, the skills evaluation based on WSM given in Formula (5) results in:

$$Default\_Skills\_Points = (0.6 * (1 + 0)) + (0.4 * (1 + 1)) = 1.4$$

Normalization of Skills Points

To ensure a standardized representation of skills points, normalization is performed. Skills points are divided by the sum of the weighted values of all technical and soft skills, and then multiplied by 100 as shown in Formula (6). Therefore, The normalization of the obtained skills points (1.4) is as follows:

$$Normalized\_Default\_Skills\_Points = (1.4 / (0.6 * 2 + 0.4 * 2)) * 100 = 70.00\%$$

Evaluation of Skills in Strict Mode:

In the Strict mode, the skills points evaluation involves the summation of a product for each skill category. Therefore, if any required skill is missing, it would result in a zero value for a specific skill category. In the given example, the job requires both Python and SQL as technical skills. However, the applicant possesses only Python skills and lacks the SQL skill. Therefore, the evaluation of technical skills in the Strict mode is:

V(Technical Skills) = V(Python) * V(SQL) = 1 * 0 = 0.

For the soft skills category, the applicant possesses the two required skills, Teamwork and Communication. Hence, the evaluation of soft skills in the Strict mode is:

V(Soft Skills) = V(Teamwork) * V(Communication) = 1 * 1 = 1

Substituting these values into the Formula (10) and considering 0.6 weight for technical skills and 0.4 weight for soft skills, the skills evaluation in the Strict mode results in:

$$Strict\_Skills\_Points = 0.6 * 0 + 0.4 * 1 = 0.4$$

Applying normalization in the Strict mode is the same as applying it in the Default mode, and it results in the following percentage:

$$Normalized\_Strict\_Skills\_Points = (0.4 / (0.6 * 2 + 0.4 * 2)) * 100 = 20.00\%$$

7.3.4 Total Score

Total Score provides an overall assessment of the applicants' fit for a particular job. The choice of the method to compute the total score depends on the desired representation of the score and the mode of compensation between criteria. In the Default mode of this matching model, a higher degree of tolerance is assumed. Therefore, the computation of the total score incorporates the use of arithmetic mean. The arithmetic mean, being a widely accepted method for calculating averages, offers a balanced approach that allows for compensation among the different criteria. Conversely, the Strict mode is designed to operate in a higher degree of sensitivity, requiring a high level of match across all criteria. In this mode, the harmonic mean is used to emphasize the importance of each criteria and discourage large compensation between them. Thus, the Strict mode ensures that all criteria are equally important in determining the total score. Using different methods for different modes allows the matching model to adapt to the desired level of tolerance and sensitivity.

Based on the normalization of education, experience, and skills points, the total score is computed as arithmetic mean and harmonic mean in the Default mode and Strict mode respectively, as shown in Formula (7) and Formula (11):

$$Total\_Score\_Default\_Mode =$$

$$(Normalized\_Education\_Points + Normalized\_Experience\_Points + Normalized\_Skills\_Points) / 3 \quad (7)$$

$$Total\_Score\_Strict\_Mode =$$

$$3 / ((1/Normalized\_Education\_Points) + (1/Normalized\_Experience\_Points) + (1/Normalized\_Skills\_Points)) \quad (11)$$

7.3.4.1 Total Score Evaluation Example

From the previous job example evaluations, the obtained normalizations are as follows:

$$Normalized\_Default\_Education\_Points = (10 - 2) / (12 - 2) * 100 = 80\%$$
$$Normalized\_Strict\_Education\_Points = (4 - 2) / (12 - 2) * 100 = 20.00\%$$

$$Normalized\_Default\_Experience\_Points = (11 - 2) / (47 - 2) * 100 = 20.00\%$$
$$Normalized\_Strict\_Experience\_Points = (4 - 2) / (47 - 2) * 100 = 4.44\%$$

$$Normalized\_Default\_Skills\_Points = (1.4 / (0.6 * 2 + 0.4 * 2)) * 100 = 70.00\%$$
$$Normalized\_Strict\_Skills\_Points = (0.4 / (0.6 * 2 + 0.4 * 2)) * 100 = 20.00\%$$

The total score in the Default mode representing the arithmetic mean in Formula (7) is then computed as follows:

$$Total\_Score\_Default\_Mode = (80 + 20 + 70) / 3 = 56.67\%$$

The total score in the Strict mode representing the harmonic mean in Formula (11) is computed as follows:

$$Total\_Score\_Strict\_Mode = 3 / ((1/20) + (1/4.44) + (1/20)) = 9.09\%$$

The arithmetic mean used in the Default mode calculates the total score as the average of the three normalizations. On the other hand, the harmonic mean used in the strict mode gives more weight to the lowest normalization. It is computed by taking the reciprocal of each normalization, summing the reciprocals, and taking the reciprocal of the sum. The harmonic mean ensures that a low normalization significantly reduces the total score. Hence, the difference in the total scores between the default and strict modes is notable. By utilizing these different approaches, the total score can be adapted to meet specific requirements and preferences.

7.3.5 Star Rating

The Star Rating is a measure that provides a quick and intuitive representation of an applicant's suitability for a job based on their total score. The star rating is determined by mapping the total score to a range of star categories, allowing for easy comparison and evaluation. In the matching model, the star rating is determined based on the total score as follows:

Star Rating:

```
0.5 star: 0-10% of the maximum Total_Score
1 star: >10-20% of the maximum Total_Score
1.5 stars: >20-30% of the maximum Total_Score
2 stars: >30-40% of the maximum Total_Score
2.5 stars: >40-50% of the maximum Total_Score
3 stars: >50-60% of the maximum Total_Score
3.5 stars: 60-70% of the maximum Total_Score
4 stars: 70-80% of the maximum Total_Score
4.5 stars: 80-90% of the maximum Total_Score
5 stars: 90-100% of the maximum Total_Score
```

7.3.5.1 Star Rating Evaluation Example

To demonstrate the application of the star rating, consider the previous example where the total score is computed in both Default and Strict modes with the following values:

$$Default\ mode\ total\ score: 56.67\%$$

$$Strict\ mode\ total\ score: 9.09\%$$

Based on these total scores, and a maximum total score of 100%, the corresponding star ratings are determined using the defined ranges as follows:

For the Default mode, the star rating is 3 stars since 56.67% falls within the range of >50-60% of the maximum total score. For the Strict mode, the star rating is 0.5 star since 9.09% falls within the range of 0-10% of the maximum total score.

In order to facilitate the implementation and application of the star rating, a function is created that maps the total score to the corresponding star category. Star rating enables employers to quickly assess and compare the suitability of applicants for a job. It offers a convenient and intuitive way to understand an applicant's overall performance and fit for the job.

7.4 Job Applicant Matching Application Implementation

7.4.1 Overview

The job applicant matching logic is implemented as Erlang/OTP Rebar3 application named "job_match[57]". This application is part of Erlang/OTP Rebar3 release called "match[58]". A release project organizes multiple top-level related applications[59] that can be deployed as executable programs. It also enables sharing common configuration[60] among these applications.

In particular, the implemented job-applicant match and evaluate application shows how Erlang can be leveraged to create complex, rule-based engines, and process semantic web data. This application integrates SERESYE and the Semantic Web Toolkit for Erlang Applications capabilities to process and evaluate BTM jobs requirements with corresponding applicants qualifications obtained in a semantic web ontology. The implementation of the application is based on the analysis of BTM jobs ontology in RDF format and its subsequent data processing in N-Triples format (obtained using a RDFLib script), a list of "iri" tuples, and a list of Erlang maps (processed by Semantic Web Toolkit for Erlang Applications). The Erlang list of maps is then processed to construct jobs facts in a list of tuples, applicants facts in a list of tuples, and then combined in one list of tuples. The combined facts represent the initial asserted KB for the developed rule engine. Matching rule processing and evaluation of jobs facts against applicants facts is developed to process the asserted KB and add matching facts to the KB. The described data manipulation and generation can be found saved in text files in the "job_match" application "/priv/data" directory[61]. The numbering in these files names indicates the order of obtaining and processing the data. These text files are written in files by the application for examination and validation purposes.

---

[57] https://github.com/MiloudEloumri/match/tree/main/apps/job_match
[58] https://github.com/MiloudEloumri/match/tree/main
[59] https://github.com/MiloudEloumri/match/tree/main/apps
[60] https://github.com/MiloudEloumri/match/blob/main/rebar.config
[61] https://github.com/MiloudEloumri/match/tree/main/apps/job_match/priv/data

7.4.2 Integration of SERESYE and Semantic Web Toolkit for Erlang Applications

The integration of SERESYE and the Semantic Web Toolkit for Erlang Application is a key aspect of this application. SERESYE is utilized to build a rule engine that operates on a KB, which contains facts about job requirements and applicant qualifications. The Semantic Web Toolkit for Erlang Application, on the other hand, is employed to process data in N-Triples format ultimately converting it into a structured list of Erlang maps.

7.4.3 Utilizing Semantic Web Ontologies

A notable feature of this application is its use of BTM jobs-applicants ontology as the dataset. This approach ensures that the knowledge operated on is well-structured and semantically rich, making the rule processing and matching logic more effective.

7.4.4 Design and Development

In addition to the common Erlang/OTP modules and files, such as rebar.cong file, the supervisor module and Application resource file[62], the matching application is structured around three main Erlang modules: "job_match.erl", "jobs_facts.erl", and "applicants_facts.erl". Each module plays a critical role in processing BTM jobs ontology data, creating the tule engine with rule matching and evaluation logic.

1. "job_match.erl": This is the core module where: (1) the BTM Jobs ontology in N-Triples format is converted by Semantic Web Toolkit for Erlang Applications into Erlang list of maps, (2) the rule engine is created and populated incorporating matching rules and evaluation functions. This module is responsible for matching each job with its applicants, and their respective evaluations based on the predefined criteria and WSM formulas introduced in the previous sections.

2. "jobs_facts.erl": This module extracts jobs facts and groups them in a list of tuples expected by the rule engine. Each job tuple is identified by the atom "job", and the job ID.

3. "applicants_facts.erl": Similarly; This module handles applicant facts and groups them in a list of tuples accepted by the rule engine. The module ensures the correct association of

[62] https://github.com/MiloudEloumri/match/tree/main/apps/job_match/src

each job with its corresponding applicants using respective job IDs. It distinguishes applicants tuples by the atom "applicant", corresponding job ID, and the applicant ID.

The used list of tuples structure ensures all jobs and applicants facts are processed. Importantly, it enables to define the logic of rule matching and evaluation for every job and applicant fact. This rule matching processing uses the power of Erlang pattern matching in functions arguments and the capabilities of SERESYE in iteratively processing the KB and firing a match rule whenever a match if found between a job and an applicant. Every matching rule processes an applicant evaluation and returns a derived matching fact that is added to the rule engine KB.

The following sections discuss these three modules in more detail.

7.4.5 Job Match Engine Module

The "job_match.erl" module represents the rule engine of the job applicant matching application. It is responsible for initializing, managing the rule engine, and evaluating the applications. The functionality of the module can be summarized as follows:

1. **Data Processing:**
   Initially, the module in its start/0 function starts the Semantic Web Toolkit for Erlang Applications which is used to process the input N-Triples data representing BTM jobs and applicants ontology. The Semantic Web Toolkit for Erlang Applications converts the data in two steps. In the first step, a stream of "iri" tuples is obtained. The next step uses the "iri" stream to generate a list of Erlang maps (stream typed). Each map holds a job or an applicant fact in its subject "S", property "P" and object "O" keys values. The list of maps is then passed to "jobs_facts.erl" and "applicants_facts.erl modules to identify, extract and construct a set of tuples facts relevant to each job or applicant, representing the initial asserted KB of the rule engine.
2. **Rule Engine Creation:**
   The module "start/0" function initializes the SERESYE "job_match" rule engine with each job and applicant fact representing the initial asserted KB used in the "match/3" rule processing.
3. **Rule Matching and Evaluation:**
   The core matching functionality of "job_match.erl" lies in its ability to match each job with corresponding applicants in the "match/3" rule. The match rule takes the rule engine name, a job fact tuple, and an applicant fact tuple as arguments. According to these arguments, it

pattern matches each job with applicants based on a job ID shared between the two tuples arguments. If a match is found, a rule is fired and the corresponding applicant is evaluated. Applicants evaluation in the match rule represent the actions of the rule. Each action is implemented in a function corresponding to a certain evaluation criteria (education, experience, and skills), as well as functions to compute the total score and star rating. Each evaluation function returns the result in a tuple structure with evaluation points and normalization values. The normalized values are then used to compute the total score which in turn is used to compute the star rating. Finally, the match rule constructs a match tuple with the obtained results and adds it to the rule engine KB.

This simple matching logic signifies the strength of Erlang pattern matching in functions arguments as well as the capabilities of SERESYE in recursively iterating the KB and identifying the matching facts.

7.4.6 Jobs Facts Module

The “jobs_facts.erl” module serves a crucial purpose in extracting and structuring each job facts in a tuple . Its main function “extract_jobs_tuples/1” receives a list of maps from the previous job match engine mode. Then, the module implementation uses a job ID and name to find and group a specific job facts. This operations relies on a map “S”, “O” and “P” keys values to identify, extract and group data in a tuple structure. Ultimately, the extract_jobs_tuples/1” function returns a structured list of tuples representing jobs facts. The process of structuring jobs data in this module can be summarized as follows:

1. **Filtering Irrelevant Data:**
   After receiving a list of maps in the “extract_jobs_tuples/1”argument, the first step in jobs data structuring involves filtering out irrelevant data. The module scans through the initial list of maps and discards any irrelevant map entries. This is achieved through “filter_unwanted_maps/1” and “is_unwanted_map/1”functions, which apply specific criteria on “P” key patterns to identify and exclude unrelated maps.

2. **Identifying Job Names:**
   After flittering out irrelevant maps, , the next step focuses on identifying maps that contain job names. This is critical as job names serve as a primary key for grouping related data. The “is_job_name_map/1” function scans each map and finds a job ID based on a map “P” key that has a value containing “requireBTMJobsTitleDP” in its IRI suffix. When the “P”

key value has this suffix, the “S” key value includes the job ID in its IRI suffix, while the “O” key value holds the job name. This is the only pattern or relation in the BTM jobs ontology that can be used to associate a job name with its job ID. All other job data can only be identified based on a job name. In other words, a job ID cannot be used to identify, extract and group other job data.

3. **Extracting and Grouping Data:**
   After isolating the job names maps, the module proceeds to extract and group related data for each job. The “extract_job_name_and_id/1” function pulls the job names and their respective IDs. These are then used by “group_by_job_name/3” to accumulate and organize related data, such as educational, experience, technical and soft skills requirements specific to each job.

4. **Structuring Data into Tuples:**
   The “convert_to_job_tuple/3” function converts the grouped data into a list of tuples. Each tuple represents a job fact including the atom ‘job’, job ID, job name, educational, experience, technical, and soft skills requirements. The atom ‘job’ is used to pattern match jobs data themselves along with the job ID, which is also used in the rules pattern matching with applicants data containing respective jobs IDS.

5. **Extracting Job Data:**
   An integral part of the process is the “extract_job_posting_data/1” function, which finds each job's specific data based on a map “P” key with various IRI suffix values related to a job requirements . It extracts values of educational level, field of study, years of experience, technical and soft skill from the grouped data though calls to “extract_value/2” function. The function then returns a structured tuple of these values.

6. **Utility Functions:**
   The module includes utility functions such as the mentioned “extract_value/2” function as well as, “proplabel/2”, and “propvalue/1” functions. These two functions are called by “extract_value/2” function to assist in extracting specific values from “O” keys of the maps based on specific “P” key suffixes indicating jobs requirements.

In short, the “jobs_facts.erl” module extracts jobs facts from maps and groups the facts in tuples. By converting the list of maps into a list of tuples, it makes it possible to process the jobs facts by the job-applicant match and evaluate engine. Erlang maps are more efficient in handling data than tuples as the maps contain keys that can be used to access their corresponding values irrespective of their order. However, SERESYE engines accept only a list of tuples as its KB.

7.4.7 Applicants Facts Module

Similar to the Jobs facts module, the “applicants_facts.erl” module plays a crucial role in identifying, extracting, and grouping each applicant data or facts in a structured list of tuples from a list of maps. Its main function “extract_applicants_tuples/1” receives a list of maps from the previous job match engine module to eventually return a list of tuples representing applicants facts or KB. To accomplish this, the module implementation uses each applicant or job seeker ID to find and group respective qualifications. An applicant ID is extracted from a map “S” key value when the map “P” key value contains “applyToJobPostingDP” suffix and a corresponding job ID is extracted from the same map “O” key value. Then, the applicant qualifications are extracted and grouped in a tuple based on the extracted applicant ID. The process of structuring applicants data can be summarized as follows:

1. **Filtering Irrelevant Data:**
   The first step involves filtering out unrelated maps. This is achieved using the “should_skip_map/1” function, which checks each map for specific “P” key patterns that should be skipped.

2. **Grouping Data by Applicant ID:**
   After filtering, the data is grouped based on each applicant’s ID found in “S” key suffix value. The “group_by_job_seeker/2” function accomplishes this by iterating over the filtered maps and aggregating them under their respective applicant IDs, which are extracted by the call to “extract_job_seeker_id/1” function. This organization is essential for the subsequent data extraction process, as it collects all relevant facts for each applicant in one place based on their IDs.

3. **Structuring Data into Tuples:**
   The next step is to convert the grouped maps into structured tuples. The “convert_to_tuple/2” function is responsible for this transformation. The function calls the “extract_job_seeker_data/1” function to retrieve specific details (name, education, experience, skills) for each applicant. Simultaneously, it extracts the job ID for each applicant using “extract_job_posting_id/1”. The identification and extraction of a job ID representing the job an applicant applied for is important because it enables easier

patterning matching with jobs facts in the rule processing. The result of this step is a structured tuple that encapsulates all the relevant facts about an applicant.

4. **Utility Functions:**

   The module includes “extract_value/2”, “proplabel/2” and “propvalue/1” utility functions which return values from “O” keys of applicants maps based on “P” keys suffixes values indicating applicants qualifications. These returned values are then used in “extract_job_seeker_data/1” function.

In short, the “applicants_facts.erl” module handles the extraction and grouping of applicants facts in a list of tuples. This is achieved through functions involving data filtering, extraction, and grouping. It returns a list of tuples representing the qualifications of job applicants. The list of tuples is then used as part of the rule engine KB.

Chapter 8

Results Discussion

8.1 Running The Job Applicant Match Application

Erlang Rebar3 tool is used for compiling and managing Erlang/OTP applications based on a certain application configuration file "rebar.config" settings. Therefore, to successfully run the job applicant match application, it must first be compiled, built, and booted using the command "rebar3 shell". This command performs several key tasks: it verifies dependencies, compiles the application along with any dependencies, and then launches an Erlang shell with the applications loaded, as shown in the following output:

```
milou@HP122021 MINGW64 ~/match (main)
   $ rebar3 shell
   ===> Verifying dependencies...
   ===> Analyzing applications...
   ===> Compiling seresye
   ===> Compiling match
   ===> Compiling semantic_relatives
   ===> Compiling job_match
   ===> Analyzing applications...
   ===> Compiling extra_apps/seresye/examples
   Eshell V11.0  (abort with ^G)
   1> ===> Booted match
   1> ===> Booted semantic
   1> ===> Booted cf
   1> ===> Booted erlware_commons
   1> ===> Booted seresye
   1> ===> Booted job_match
   1> ===> Booted semantic_relatives
   1> ===> Booted sasl
   1>
```

From this output, it is evident that the "job_match" application and its dependencies, including the critical "semantic" and "seresye" applications, are successfully built, compiled, and booted without any errors. Once the build and compilation process are complete, the "job_match" rule engine can be started using the command "job_match:start()." The expected output of this command is as follows:

```
1> job_match:start().
ok
2>
```

The output "ok" signifies that the “job_match” rule engine is started successfully without any errors. Upon running this command, the rule engine is created and populated. This step is critical as it ensures that the constructed KB and defined match rules are processed, resulting in deriving and adding new facts to the rule engine KB.

8.2 Querying The Rule Engine Results

1. Getting All Facts

A central aspect of evaluating the performance of the application lies in querying the results of the rule engine KB. This is achieved through the SERESYE function “seresye_srv:get_kb/1”, which is designed to retrieve all facts processed by the rule engine. This function takes the name of the rule engine as its argument. When the command “seresye_srv:get_kb(job_match).” is executed, it returns a list of tuples representing all the facts in the KB. The following shows an example of the output :

```
2> seresye_srv:get_kb(job_match).
[{match,<<"JPosting22">>,<<"JSeeker126">>,
        {<<"DataScientistJuniorDataScienceAnalytics">>,
         <<"MohamadLucero">>,
         {score,51.11},
         {stars,3.0},
         {education,8,60.0},
         {experience,8.0,13.33},
         {skills,1.6,80.0}}},
 {applicant,<<"JPosting22">>,
            {<<"JSeeker126">>,
             {<<"MohamadLucero">>,
              {education,<<"Bachelor">>,<<"Computing">>},
              {experience,2,
<<"MethodologiesAndTechniquesUsedAsAJuniorDataScientist">>,
                           <<"DataAnalysisExperience">>},
{techskills,<<"KnowledgeAndExperienceOfLargeComplexDataAnalyticsOrIntel
ligenceP"...>>,
<<"StatisticalAndPatternRecognitionSkills">>},
              {softskills,<<"Numeracy">>,<<"ProblemSolving">>}}}},
 {match,<<"JPosting11">>,<<"JSeeker63">>,
        {<<"DigitalSecurityAuditor">>,<<"ImaanKendall">>,
         {score,69.57},
         {stars,3.5},
         {education,5,50.0},
         {experience,34.0,58.72},
         {skills,2.0,100.0}}},
 {...}|...]
3>
```

In this output, the presence of "match" tuples along with their comprehensive evaluation data (scores, stars, education, experience, and skills) is a clear indication that the matching rules are successfully fired and added to the KB. Each "match" tuple associates a job ID with an applicant and provides a detailed evaluation of the applicant's suitability for the job based on the established criteria and the WSM formulas. The accuracy and relevance of these matches and evaluations are determined by comparing the results with the original BTM jobs ontology data.

All KB facts are saved in a file[63] for validation purposes. Moreover, the results of data processing, transformation and construction are also saved in files in the application "/prv/data" directory[64].

The notations such as (<<"JPosting22">>) represent Erlang binary data. Erlang binary data is more efficient in terms of space and performance than other data types such as atoms (literals). Erlang best practice discourages the use of a large number of atoms especially when created dynamically, as atoms are not auto garbage collected by the Erlang Virtual Machine (VM).

2. Getting Specific Facts

In addition to retrieving all facts from the rule engine KB, specific fact templates can be used to obtain results about individual matches and evaluations. This feature allows for targeted analysis of the results, enabling a deeper understanding of the matching logic and results. For instance, to determine the match and corresponding evaluation between a specific job posting "JPosting1" and a job seeker "JSeeker1", a specific fact query is structured as follows:

```
seresye_srv:query_kb(job_match, {match, <<"JPosting1">>,
<<"JSeeker1">>, '_'}).
```

In the "seresye_srv:query_kb/2" function, the first argument specifies the engine name "job_match", while the second argument is a match tuple representing the fact template to be matched. The atom '_' acts as a wildcard allowing flexible querying. Executing this query, the result shows the evaluation of "JSeeker1":

```
3> seresye_srv:query_kb(job_match, {match, <<"JPosting1">>,
<<"JSeeker1">> , '_'}).
[{match,<<"JPosting1">>,<<"JSeeker1">>,
        {<<"DigitalSecurityManagerOfficer">>,<<"AustinLeigh">>,
         {score,64.81},
         {stars,3.5},
         {education,7,35.71},
         {experience,34.0,58.72},
         {skills,2.0,100.0}}}]
4>
```

[63] https://github.com/MiloudEloumri/match/blob/main/apps/job_match/priv/data/9-jobs-applicants-asserted-inferred-kb.txt
[64] https://github.com/MiloudEloumri/match/tree/main/apps/job_match/priv/data

This output reveals a successful match between “JPosting1” and “JSeeker1”, providing a detailed evaluation of the seeker's qualifications. The actual BTM jobs ontology data and the application of the WSM formulas confirm the accuracy and relevance of this match. Furthermore, the specificity of this query can be demonstrated by changing the job posting ID or job seeker ID. For example, querying a non-existent match between “JPosting2” and “JSeeker1” results in an empty list:

```
4> seresye_srv:query_kb(job_match, {match, <<"JPosting2">>,
<<"JSeeker1">> , '_'}).
[]
5>
```

This indicates that “JSeeker1” did not apply for “JPosting2”, verifying the application's precision in matching jobs with applicants. Moreover, the absence of a match in such scenarios validates the integrity of the rule engine's logic and the construction of the KB. The ability to query specific facts demonstrates not only the flexibility and precision of the rule engine but also ensures transparency and traceability in the matching and evaluation results.

8.3 Getting Facts using 'fun' for Fine-Grained Queries

The use of Erlang's 'fun' adds powerful querying capabilities. This feature allows for the construction of complex, fine-grained fact templates, enhancing the querying capabilities of the SERESYE rule engines. A 'fun' in Erlang is an anonymous function that can be used as an element in a tuple or as an argument for other functions. This function returns a Boolean value indicating whether the element matches the specified fact template. The use of 'fun' enables the extraction of highly specific facts. For instance, to query evaluations of both “JSeeker1” and “JSeeker3” in “JPosting1”, the following 'fun' is used:

```
seresye_srv:query_kb(job_match, {match, <<"JPosting1">>, fun (X) -> (X
== <<"JSeeker1">>) or (X == <<"JSeeker3">>) end, '_'}).
```

Executing this query, the result is given as follows:

```
5> seresye_srv:query_kb(job_match, {match, <<"JPosting1">>, fun (X) ->
(X == <<"JSeeker1">>) or (X == <<"JSeeker3">>) end, '_'}).
[{match,<<"JPosting1">>,<<"JSeeker3">>,
        {<<"DigitalSecurityManagerOfficer">>,<<"SteffanBeattie">>,
         {score,36.72},
         {stars,2.0},
         {education,10,57.14},
         {experience,20.0,33.03},
         {skills,0.4,20.0}}},
 {match,<<"JPosting1">>,<<"JSeeker1">>,
        {<<"DigitalSecurityManagerOfficer">>,<<"AustinLeigh">>,
         {score,64.81},
```

```
          {stars,3.5},
          {education,7,35.71},
          {experience,34.0,58.72},
          {skills,2.0,100.0}}}]
6>
```

This demonstrates how 'fun' can be used to filter and retrieve specific matches from the KB, offering a refined view of the evaluation results. Moreover, 'fun' can be leveraged to build even more specific queries. For example, to find applicants of “JPosting1” with a score greater than 50, the following query is used:

```
seresye_srv:query_kb(job_match,
    {match, <<"JPosting1">>, '_',
        fun({_JobTitle, _ApplicantName, {score, Score}, _, _, _, _}) ->
            Score > 50
        end
    }
).
```

The query result is:

```
6> seresye_srv:query_kb(job_match,
    {match, <<"JPosting1">>, '_',
        fun({_JobTitle, _ApplicantName, {score, Score}, _, _, _, _}) ->
            Score > 50
        end
    }
).
[{match,<<"JPosting1">>,<<"JSeeker1">>,
        {<<"DigitalSecurityManagerOfficer">>,<<"AustinLeigh">>,
         {score,64.81},
         {stars,3.5},
         {education,7,35.71},
         {experience,34.0,58.72},
         {skills,2.0,100.0}}}]
7>
```

This query returns only the applicants (“JSeeker1” in this case) who meet the specified score criterion. As another example, 'fun' can be used to confirm the absence of certain facts. For instance, querying “JPosting1” applicants with a star rating greater than 4 results in an empty list, indicating that no applicant for “JPosting1” has such a high star rating:

```
7> seresye_srv:query_kb(job_match,

    {match, <<"JPosting1">>, '_',

        fun({_JobTitle, _ApplicantName, _, {stars, Stars}, _, _, _}) ->

            Stars > 4

        end
```

```
    }
).
[]
8>
```

These examples highlight the powerful and flexible nature of 'fun' in the Erlang environment, especially when applied to complex data structures. The ability to construct such detailed queries is invaluable in validating the application's logic and ensuring the accuracy of its rule processing. Likewise, all jobs can be queried for applicants with star rating greater than 4 as follows:

```
8> seresye_srv:query_kb(job_match,
    {match, '_', '_',
        fun({_JobTitle, _ApplicantName, _, {stars, Stars}, _, _, _}) ->
            Stars > 4
        end
    }
).
[{match,<<"JPosting19">>,<<"JSeeker107">>,
        {<<"FinancialServicesITGovernanceRiskAndComplianceManager">>,
         <<"LauraObrien">>,
         {score,86.67},
         {stars,4.5},
         {education,8,60.0},
         {experience,49.0,100.0},
         {skills,2.0,100.0}}},
 {match,<<"JPosting9">>,<<"JSeeker50">>,
        {<<"FinancialServicesEnterpriseArchitect">>,<<"GuyBenson">>,
         {score,86.67},
         {stars,4.5},
         {education,8,60.0},
         {experience,49.0,100.0},
         {skills,2.0,100.0}}},
 {match,<<"JPosting14">>,<<"JSeeker84">>,
        {<<"FinancialServicesCloudServicesManager">>,
         <<"JameelNielsen">>,
         {score,86.67},
         {stars,4.5},
         {education,8,60.0},
         {experience,49.0,100.0},
         {skills,2.0,100.0}}},
 {match,<<"JPosting9">>,<<"JSeeker46">>,
        {<<"FinancialServicesEnterpriseArchitect">>,
         <<"JoyceSheldon">>,
         {score,86.67},
         {stars,4.5},
         {education,8,60.0},
         {experience,49.0,100.0},
         {skills,2.0,100.0}}},
 {match,<<"JPosting14">>,<<"JSeeker81">>,
        {<<"FinancialServicesCloudServicesManager">>,
         <<"TabathaBarnett">>,
```

```
         {score,86.67},
         {stars,4.5},
         {education,8,60.0},
         {experience,49.0,100.0},
         {skills,2.0,100.0}}},
 {match,<<"JPosting19">>,<<"JSeeker105">>,
        {<<"FinancialServicesITGovernanceRiskAndComplianceManager">>,
         <<"SashaWooten">>,
         {score,86.67},
         {stars,4.5},
         {education,8,60.0},
         {experience,49.0,100.0},
         {skills,2.0,100.0}}},
 {match,<<"JPosting14">>,<<"JSeeker79">>,
        {<<"FinancialServicesCloudServicesManager">>,
         <<"ShauryaMack">>,
         {score,86.67},
         {stars,4.5},
         {education,8,60.0},
         {experience,49.0,100.0},
         {skills,2.0,100.0}}},
 {match,<<"JPosting14">>,<<"JSeeker82">>,
        {<<"FinancialServicesCloudServicesManager">>,
         <<"FergusOSullivan">>,
         {score,86.67},
         {stars,4.5},
         {education,8,60.0},
         {experience,49.0,100.0},
         {skills,2.0,100.0}}}]
9>
```

8.4 Testing Results using Erlang EUnit Tests

Unit tests involve testing components of an application independently from the rest of the application. Erlang EUnit[65] is a powerful and flexible testing framework for writing and running unit tests. It provides a way to validate the functionality of individual functions or modules.

The job-applicant matching and evaluation application includes "job_match_tests module[66] that includes simple Eunit tests. It uses the EUnit framework to validate the matching and evaluation of applicants based on the constructed KB and the defined matching rules. The module includes the following key functions:

setup/0: Initializes the "seresye" engine and adds the rules from the "job_match" module.

test_applicants/0: Provides sample of jobs and applicants facts to validate against the matching and evaluation assertions.

---

[65] https://www.erlang.org/doc/apps/eunit/chapter.html
[66] https://github.com/MiloudEloumri/match/blob/main/apps/job_match/test/job_match_tests.erl

rules_test/0: It is added to present how multiple tests can be grouped in one test function by asserting the sample data and validating the matches for multiple applicants.

Individual Test Functions: "test_match_seeker1/0", "test_match_seeker2/0", and "test_match_seeker3/0" test the matching and evaluation logic for specific applicants (JSeeker1, JSeeker2, and JSeeker3) separately.

EUnit Test Generator: "job_match_test_/0" collects all individual test functions and runs them as part of the EUnit test suite resulting in 5 tests including job_match_test_ itself.

These tests can be run using the following command:

```
rebar3 eunit --module=job_match_tests
```

This command compiles the necessary modules and execute the tests, as follows:

```
milou@HP122021 MINGW64 ~/match (main)
$ rebar3 eunit --module=job_match_tests
===> Verifying dependencies...
===> Analyzing applications...
===> Compiling seresye
===> Compiling match
===> Compiling semantic_relatives
===> Compiling job_match
===> Analyzing applications...
===> Compiling extra_apps/seresye/examples
===> Performing EUnit tests...
======================= EUnit ========================
module 'job_match_tests'
  job_match_tests: rules_test...[0.062 s] ok
  job_match_tests:237: job_match_test_...ok
  job_match_tests:238: job_match_test_...ok
  job_match_tests:239: job_match_test_...ok
  job_match_tests:240: job_match_test_...ok
  [done in 0.140 s]
=======================================================
  All 5 tests passed.

milou@HP122021 MINGW64 ~/match (main)
$
```

The "job_match_tests" module serves as a template for further and more comprehensive testing. The match assertions used in the module are taken directly from the results of the rule engine and are not verified separately or manually. For robust testing, expected data should be calculated or verified independently to ensure the accuracy of the assertions. Moreover, for comprehensive testing of results and evaluation of performance, several Erlang testing tools can be used including Erlang timer Module, Erlang observer Application, Common Test, Exometer and Erlang Performance Lab.

8.5 Results Observation and Enhancement

The implementation of the job-applicant matching and evaluation application returned insightful results, particularly in the processing and structuring of the BTM jobs ontology data and the creation of the rule engine that matches and evaluates applicants.

1. Job Facts Processing:

The results demonstrate that all of the 27 job postings instances present in the BTM jobs ontology are successfully processed and structured as expected[67]. This proves the application's capability to accurately interpret and organize job-related data.

2. Applicants Facts Processing :

Although all 160 applicant or job seeker instances from the BTM jobs ontology are processed successfully, it is observed the presences of additional "undefined" tuples within the applicants' facts[68]. While these unexpected tuples do not impact the overall matching results, they do highlight a need for further examination, debugging and improvement.

3. Matching Rule Facts Processing:

Corresponding to the 160 applicants in the BTM jobs ontology, the matching rule processing resulted in 160 derived "match" facts[69]. This alignment confirms the effectiveness of the rule processing in evaluating applicants qualifications against jobs requirements.

4. Updating SERESYE to handle Erlang maps:

One of the significant challenges faced in the implementation is related to the data type limitation of SERESYE. SERESYE's supports only tuple data type as its KB facts. This limitation introduced complexities, especially considering the Semantic Web Toolkit for Erlang Applications processes N-Triples data into a list of Erlang maps. This list of maps then requires processing to identify, extract and group jobs and applicants facts in a list of tuples expected by SERESYE. Erlang maps, with their key-value structure, offer a more efficient means of handling and accessing specific data.

---

[67] https://github.com/MiloudEloumri/match/blob/main/apps/job_match/priv/data/6-jobs-asserted-facts-tuples.txt

[68] https://github.com/MiloudEloumri/match/blob/main/apps/job_match/priv/data/7-applicants-asserted-facts-tuples.txt

[69] https://github.com/MiloudEloumri/match/blob/main/apps/job_match/priv/data/9-jobs-applicants-asserted-inferred-kb.txt

Updating SERESYE to handle Erlang maps directly could significantly enhance the efficiency and flexibility of its rule engines.

In summary, while the job applicant matching application successfully processes and matches job postings and applicants with their corresponding evaluations, the observation of additional "undefined" tuples suggests areas for improvement. Addressing these inconsistencies and enhancing SERESYE to handle Erlang maps directly could greatly optimize the application's performance and reliability.

8.6 Answering Research Questions and Meeting Objectives

The research questions and objectives outlined in Chapter 1 are answered and discussed in detail throughout the thesis content. This section synthesizes the answers to these questions and demonstrates how the objectives are met as follows:

Objective 1: Examine Current Personnel Selection Methods

Research Question 1: What are the key challenges and limitations in the current personnel selection methods?

To address this objective and research question, the thesis critically examined the state of the art in personnel selection methods. Accordingly, the thesis identified several key challenges and limitations in current personnel selection methods, including biases, high costs, inefficiencies in traditional methods, and the inability to process multiple criteria simultaneously. This emphasizes the need for advanced systems that combine human expertise with technological solutions to effectively address these issues.

These challenges and limitations are detailed in several sections, including the following: Section 1.1.2 Recruitment and Selection in Modern Human Resource Management emphasized the need to address challenges such as unreasonable and unrealistic job requirements analysis, discrimination in employment standards, inappropriate recruitment methods, long recruitment and selection cycle, and difficulties in attracting candidates. Section 2.4.3 Selection Process as a Decision-Making Process discussed how decision-making complexities impact the selection process. Section 2.5 Challenges in the Recruitment and Selection Process identified key challenges, such as biases, high costs, and inefficiencies in traditional methods. Section 3.7 Expert Systems Applications in Personnel Selection explored the application of expert systems in overcoming some of these limitations. Section 5.7 Multi-Criteria Decision Making Approaches in Personnel Selection analyzed how MCDM can provide structured and unbiased selection processes, addressing some of the identified challenges.

Objective 2: Analyze BTM Jobs-Applicant Ontology

Research Question 2: How can BTM jobs-applicants ontology be effectively analyzed and used as dataset in the job-applicant matching and evaluation application?

Research Question 2 is addressed through the analysis of the BTM jobs ontology stored in RDF format using Protégé ontology editor. The BTM jobs dataset and its analysis are discussed in detail in the research methodology provided in Chapter 6, especially in Section 6.2. Examining the ontology concepts and their relationships enables identifying both matching and evaluation patterns and criteria. As a result, the BTM ontology was successfully used as a dataset in the job-applicant matching and evaluation application.

Objective 3: Study SERESYE and Semantic Web Toolkit for Erlang Application Integration

Research Question 3: How can SERESYE and Semantic Web Toolkit for Erlang Applications be applied to enhance the job-applicant matching and evaluation process?

The study found that efficient integration of SERESYE and the Semantic Web Toolkit for Erlang Applications required preprocessing and transformation of RDF into NT format. This was achieved through the use of RDFLib. This integration was detailed in several sections, including the following: Section 6.2.4 BTM Jobs Ontology: Data Preprocessing Using RDFLib explained the use of RDFLib and the process of obtaining ontology data in NT format. Section 6.2.5 BTM Jobs Ontology: Data Processing Using Semantic Web Toolkit for Erlang Applications examined how semantic web data can be processed in Erlang and integrated with SERESYE. Section 6.2.6 Constructing Rule Engine Knowledge Base: Jobs and Applicants Facts detailed the KB construction using SERESYE. Section 6.2.7 Implementing the Matching Rules discussed the implementation concepts of matching rules. Then, Chapter 7 detailed the procedure and the practical implementation and integration of these technologies. The study found that these tools significantly enhance the job-applicant matching process by enabling efficient rule processing and comprehensive knowledge representation.

Objective 4: Design and Implement a Matching and Evaluation Model

Research Question 4: How can an effective job-applicant matching model be designed and implemented based on the integration of SERESYE, Semantic Web Toolkit for Erlang Applications, and MCDM using WSM?

The design and development concepts and stages of the matching application were discussed in detail in the Methodology chapter. Then, Chapter 7 provided a comprehensive overview of the model design and the practical implementation. The key concepts in the design and development lie on the ontology data analysis and subsequent processing, KB constructing, matching rules pattern identification incorporating WSM evaluations as the rules actions. Consequently, an effective job-applicant matching and evaluation model was designed and implemented.

Objective 5: Evaluate Results

Research Question 5: What are the results findings and limitations?

As presented in this chapter, the results of the developed matching model revealed that it accurately matched all dataset instances represented by 27 jobs and 160 applicants with respective evaluations. However, some limitations were identified, such as the presence of additional "undefined" tuples in the applicant facts and the need for updating SERESYE to handle Erlang maps directly and the necessity for incorporating quantitative testing metrics to evaluate the application's performance comprehensively. These findings highlight areas for further improvement and optimization.

Objective 6: State Findings, Discuss Limitations, and Identify Future Work

Research Question 6: What are the potential areas of improvement and future directions for the developed matching model?

The research discussed encountered limitations including inconsistencies in processing applicant facts, evaluation criteria limitations and the limitation of SERESYE. Moreover, the research identified future improvements including developing a web interface for the matching system and extending its functionalities to include the implementation of the proposed strict mode evaluation and the related operations such as emailing and scheduling interviews. Integrating quantitative testing tools will further enhance the robustness and reliability of the developed model. These limitations and future work are discussed in detail in the following Conclusion chapter.

Chapter 9

Conclusion

9.1 Summary

Various topics and technologies related to the thesis topic were introduced and discussed. The thesis addresses the problem of personnel selection by developing jobs and applicants matching and evaluation application based on Erlang, semantic web ontologies represented by BTM jobs ontology and MCDM WSM evaluation formulas. The thesis reviewed HRM practices and processes with a focus on its selection process. It then introduced Erlang programming language and its Expert systems especially ERESYE and its enhanced version SERESYE, as well as the Semantic Web Toolkit for Erlang Applications with a literature review on the application of expert system in personnel selection. A brief background on the semantic web technologies focusing on its ontology concepts and application is likewise discussed. Moreover, the thesis provided a comprehensive background on MCDM concepts and methods, as well as a literature review on the application of MCDM methods in personnel selection. Then, the foundation and implementation of the job-applicant matching and evaluation application is discussed in detail.

The developed job-applicant matching and evaluation application represents a significant advancement in the realm of personnel selection problem. Developed as Erlang/OTP, Rebar3 application, it integrates two crucial Erlang technologies: SERESYE, and the Semantic Web Toolkit for Erlang Application. These technologies are used to construct a sophisticated rule-based engine capable of processing complex semantic data represented by BTM jobs ontology modeling 27 jobs requirements and 160 applicants qualifications.

At its core, the job-applicant matching and evaluation application is designed to match and evaluate applicants qualification based on several criteria related to jobs education, experience, and skills requirements. The evaluation process is based on the use of WM incorporated in several functions representing the rule matching actions. The development process involves analyzing BTM jobs ontology and several sophisticated subsequent data processing, transformation, extraction and structuring. Three main modules constitute the development process: “job_match.erl”, “jobs_facts.erl”, and “applicants_facts.erl”.

The ”job_match.erl” module serves as the heart of the application, where the rule engine is created and managed. It processes the initial constructed KB facts and then matches jobs with corresponding applicants, ultimately deriving and adding new matching facts to the KB, representing the evaluation of applicants based on the defined criteria. The “jobs_facts.erl” and “applicants_facts.erl” modules are responsible for structuring initial jobs and applicants facts respectively. They thoroughly extract, filter, and organize the facts into tuples that are then processed by the rule engine.

The application's innovative use of semantic web ontologies as its dataset ensures that the data is well-structured and semantically rich. This approach enhances knowledge representation as well as matching and evaluation results. Moreover. The application demonstrates the capabilities of Erlang in pattern matching, the power of SERESYE in rule processing and the advanced features of Semantic Web Toolkit for Erlang Application in handing sematic web ontologies . It stands out as a robust solution in the field of personnel selection.

The results of the application demonstrate the feasibility of integrating SERESYE rule processing with semantic web datasets, leading to enhanced job-applicant matching and evaluation results. This thesis thus makes a significant contribution to personnel selection, expert systems, semantic web, and decision-making systems, providing a foundation for future advancements.

9.2 Limitations

Due to time constraints, certain limitations exist in the current implementation of the job-applicant matching and evaluation application. These limitations highlight areas for future development and refinement:

1. Strict Mode of Applicant Evaluation:

The planned implementation of the Strict mode for evaluating applicants remains incomplete. This mode aims to provide an alternative and rigorous evaluation method in addition to the implemented Default mode. The complete implementation of the Strict mode is planned in the next application update. Implementing this feature will enhance the application's capability to examine applicants against more rigorous evaluation.

2. Inconsistencies in Applicant Facts:

The observed inconsistencies in the constructed applicants' facts represent a limitation that demands consideration. Addressing these discrepancies will improve the reliability of the application's data processing. Additionally, the proposed enhancement of SERESYE to process Erlang maps directly as part of its KB is another critical update that could significantly simplify the creation of rule engines operating on maps data structure.

3. Evaluation Criteria Limitations:

Currently, the application focuses on essential evaluation criteria, namely education, experience, and skills, as well as the structure and concepts of the used BTM jobs ontology. This specific implementation indicates that adapting the application to a different jobs ontology would require certain implementation modifications.

4. Testing and Verification

The testing and verification of the results are mainly qualitative and based on observations and examining generated results in text files. Furthermore, a simple EUnit test module is implemented serving as a template for more rigorous testing. While this approach ensures the application's logic is sound, it is essential to incorporate advanced quantitative metrics such as accuracy, scalability, and performance to provide a more comprehensive assessment. Future work should consider using metrics such as F-Score, a common metric in classification problems, to evaluate the precision and recall of the matching process. In addition to EUnit testing, Erlang provides several testing and performance analysis tools and technologies that can be integrated into the application to perform various tests. Key tools include:

Erlang timer Module: Provides timing functions to measure execution time.

Erlang observer Application: Enables graphical observation of Erlang system characteristics, including system load, memory usage, and process information.

Common Test: Used for automated testing of Erlang programs.

Exometer: A performance monitoring library that collects metrics on response times, throughputs, and resource utilization.

Erlang Performance Lab: Used to analyze the performance of Erlang applications.

Despite these limitations, the application lays a robust foundation for applicants matching and evaluation and it can be adapted and extended in several ways. The current focus on essential evaluation criteria provides a solid base for incorporating additional factors. Criteria such as language proficiency, certifications, and training can be similarly integrated to create a more comprehensive applicant evaluation application. Similarly, the application's foundational structure and logic offer a valuable template for developing similar applications in the personnel selection domain.

In short, while the job applicant matching application demonstrates promising capabilities, its current version is marked by certain limitations due to time constraints. Future updates are expected to address these limitations, enhancing the application's functionality and adaptability. The potential for expanding the application's criteria and adapting it to various ontologies opens possibilities for broader applications and continued development in the field.

9.3 Future Work

The scope of future work involves the following:

1. Addressing Current Limitations:

The primary focus for future work will be on resolving the limitations identified in the current version of the job applicant matching application. This includes the implementation of the Strict mode of applicant evaluation, resolving inconsistencies in applicants' facts, enhancing SERESYE to process Erlang maps, and incorporating quantitative testing metrics.

2. Integration with Erlang Mnesia Database:

To improve the functionality and flexibility of the matching application, a new module is planned for development. This module will enable the storage of the rule engine's KB in the Erlang Mnesia database. Such integration not only extends the application's capabilities but also lays the basis for future integration with other Erlang-based technologies and applications.

3. Web Interface Development:

Creating a web interface using Erlang Zotonic web framework, for example, is another key area of future development. This interface will facilitate the online accessibility of the application, allowing organizations to input their job and applicant ontology files and conveniently query results. The user-friendly web interface will significantly enhance the usability and reach of the application.

4. Extension to Selection Process:

Expanding the application's scope to include other aspects of the selection process based on the web interface is also envisioned. This expansion could incorporate functionalities such as emailing candidates, scheduling interviews, and extending job offers, thereby making the application a comprehensive tool for recruitment and selection management.

5. Implementation of Multiple Parallel SERESYE Rule Engines:

Exploring the use of multiple parallel SERESYE rule engines, each with its KB, represents an innovative approach to processing and matching. This could involve one rule engine for handling job data, another for processing applicant data, and a third that combines these data to perform matching and evaluation operations. Such structure would enable testing SERESYE multiple parallel rule engines and their interactions.

6. Leveraging Poolboy for Concurrent Processes:

Given the complexity of managing parallel processes and the potential size of ontologies, integrating the Poolboy library is proposed. Poolboy[70] is an Erlang-based lightweight, generic pooling library designed for efficient concurrent processes management. It offers features such as concurrency level control, process lifecycle management, and overload protection, which are crucial when dealing with parallel processes. Moreover, the use of Poolboy will facilitate distributing the processing of large ontologies data across multiple concurrent processes. Each process in the Poolboy-managed pool could handle a segment of the ontology data, thereby enhancing performance and scalability. Implementing the features provided by Poolboy directly using OTP can be complex and time consuming. Poolboy simplifies[71] these complexities, providing an accessible and efficient means of leveraging Erlang/OTP concurrent process capabilities.

In short, the future enhancement of the job applicant matching application encompasses both technical improvements and functional expansions. Addressing current limitations, integrating with advanced Erlang tools, and expanding its capabilities to encompass wider aspects of selection process, the application can continue to evolve into a more robust, flexible, and user-friendly tool. These developments will not only refine the application's performance but also extend its applicability and utility in the HRM recruitment and selection processes.

[70] https://github.com/devinus/poolboy
[71] https://elixirschool.com/en/lessons/misc/poolboy

References

[1] "Human Resource Management V2," Nicolet College. Accessed: Jul. 02, 2023. [Online]. Available: https://nicoletcollege.pressbooks.pub/humanresourcemanagementv2/

[2] G. Ye, "Critically Discuss Challenges and Recommendations in Recruitment and Selection," in *Proceedings of the 2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022)*, Atlantis Press, 2022, pp. 274–279. doi: 10.2991/aebmr.k.220307.043.

[3] M. Khorami and R. Ehsani, "Application of Multi Criteria Decision Making approaches for personnel selection problem: A survey," *Journal of Engineering Research and Applications*, vol. 5, no. 5, pp. 14–29, 2015, Accessed: Nov. 03, 2023. [Online]. Available: www.ijera.com

[4] P. Achten, M. van Eekelen, P. Koopman, and M. T. Morazán, "Trends in Trends in Functional Programming 1999/2000 versus 2007/2008," *Higher-Order and Symbolic Computation*, vol. 23, no. 4, pp. 465–487, Nov. 2010, doi: 10.1007/s10990-011-9074-z.

[5] J. Backus, "Can programming be liberated from the von Neumann style?: a functional style and its algebra of programs," *Commun ACM*, vol. 21, no. 8, pp. 613–641, 1978, doi: 10.1145/359576.359579.

[6] K. Hammond, "Why Parallel Functional Programming Matters: Panel Statement," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6652 LNCS, no. March, 2011, pp. 201–205. doi: 10.1007/978-3-642-21338-0_17.

[7] M. Vujošević-Janičić and D. Tošić, "The Role of Programming Paradigms in the First Programming Courses," *The Teaching of Mathematics*, vol. 11, no. 2, pp. 63–83, 2008.

[8] J. Hughes, "Why Functional Programming Matters," *Comput J*, vol. 32, no. 2, pp. 98–107, Feb. 1989, doi: 10.1093/comjnl/32.2.98.

[9] J. Hunt, "Functional Programming," in *A Beginner's Guide to Scala, Object Orientation and Functional Programming*, Cham: Springer International Publishing, 2014, pp. 33–37. doi: 10.1007/978-3-319-06776-6_5.

[10] M. McDonnell, "Functional Programming," in *Quick Clojure*, Berkeley, CA: Apress, 2017, pp. 29–43. doi: 10.1007/978-1-4842-2952-1_3.

[11] V. Torra, "An Introduction to Functional Programming Languages," in *Scala: From a Functional Programming Perspective*, vol. 9980, in Lecture Notes in Computer Science,

vol. 9980. , Cham: Springer International Publishing, 2016, pp. 1–6. doi: 10.1007/978-3-319-46481-7_1.

[12] J. Armstrong, “Erlang,” *Commun ACM*, vol. 53, no. 9, pp. 68–75, Sep. 2010, doi: 10.1145/1810891.1810910.

[13] S. Vinoski, “Concurrency with Erlang,” *IEEE Internet Comput*, vol. 11, no. 5, pp. 90–93, Sep. 2007, doi: 10.1109/MIC.2007.104.

[14] Á. F. Díaz, C. B. Earle, and L.-Å. Fredlund, “eJason: An Implementation of Jason in Erlang,” in *Programming Multi-Agent Systems. ProMAS 2012*, vol. 7837, Springer, Berlin, Heidelberg, 2013, pp. 1–16. doi: 10.1007/978-3-642-38700-5_1.

[15] I. Valkov, N. Chechina, and P. Trinder, “Comparing languages for engineering server software,” in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing - SAC ’18*, New York, New York, USA: ACM Press, 2018, pp. 218–225. doi: 10.1145/3167132.3167144.

[16] A. Di Stefano and C. Santoro, “eXAT: an Experimental Tool for Programming Multi-Agent Systems in Erlang,” *In 4th AI*IA/TABOO Joint Workshop on Objects and Agents (WOA 2003)*, pp. 121–127, 2003, [Online]. Available: http://www.lia.deis.unibo.it/books/woa2003/pdf/09.pdf

[17] C. Santoro, “eXAT : Software Agents in Erlang,” in *Erlang/OTP User Conference 2005*, 2005. Accessed: Feb. 09, 2017. [Online]. Available: http://erlang.org/euc/05/Santoro.pdf

[18] A. Di Stefano and C. Santoro, “Designing collaborative agents with eXAT,” in *13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, IEEE Comput. Soc, 2004, pp. 15–20. doi: 10.1109/ENABL.2004.23.

[19] A. Di Stefano and C. Santoro, “Building Semantic Agents in eXAT.,” *In AI*IA/TABOO Joint Workshop on Objects and Agents (WOA 2005)*, pp. 28–36, 2005, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.610&rep=rep1&type=pdf

[20] A. Di Stefano and C. Santoro, “Using the Erlang Language for Multi-Agent Systems Implementation,” in *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, IEEE, 2005, pp. 679–685. doi: 10.1109/IAT.2005.141.

[21] Á. Fernández Díaz, C. Benac Earle, and L.-Å. Fredlund, “Erlang as an implementation platform for BDI languages,” in *Proceedings of the eleventh ACM SIGPLAN workshop on Erlang workshop - Erlang ’12*, New York, New York, USA: ACM Press, 2012, p. 1. doi: 10.1145/2364489.2364491.

[22] W. Turek *et al.*, “Highly scalable Erlang framework for agent-based metaheuristic computing,” *J Comput Sci*, vol. 17, pp. 234–248, Nov. 2016, doi: 10.1016/j.jocs.2016.03.003.

[23] D. Krzywicki, W. Turek, A. Byrski, and M. Kisiel-Dorohinicki, “Massively concurrent agent-based evolutionary computing,” *J Comput Sci*, vol. 11, pp. 153–162, Nov. 2015, doi: 10.1016/j.jocs.2015.07.003.

[24] D. Krzywicki *et al.*, “Generation-free Agent-based Evolutionary Computing,” *Procedia Comput Sci*, vol. 29, pp. 1068–1077, 2014, doi: 10.1016/j.procs.2014.05.096.

[25] C. Varela, C. Abalde, L. Castro, and J. Gulías, “On modelling agent systems with erlang,” in *Proceedings of the 2004 ACM SIGPLAN workshop on Erlang - ERLANG ’04*, New York, New York, USA: ACM Press, 2004, pp. 65–70. doi: 10.1145/1022471.1022481.

[26] M. Wooldridge and N. R. Jennings, “Agent theories, architectures, and languages: A survey,” in *Wooldridge M.J., Jennings N.R. (eds) Intelligent Agents*, Springer, Berlin, Heidelberg, 1995, pp. 1–39. doi: 10.1007/3-540-58855-8_1.

[27] S. Virdhagriswaran, D. Osisek, and P. O’Connor, “Standardizing agent technology,” *StandardView*, vol. 3, no. 3, pp. 96–101, Sep. 1995, doi: 10.1145/226191.226200.

[28] M. Wooldridge, “Intelligent Agents: The Key Concepts,” in *Multi-Agent Systems and Applications II. ACAI 2001. Lecture Notes in Computer Science*, vol. 2322, Springer, Berlin, Heidelberg, 2002, pp. 3–43. doi: 10.1007/3-540-45982-0_1.

[29] M. Wooldridge, “Agent-based software engineering,” *IEE Proceedings - Software Engineering*, vol. 144, no. 1, p. 26, 1997, doi: 10.1049/ip-sen:19971026.

[30] S. Franklin and A. Graesser, “Is It an agent, or just a program?: A taxonomy for autonomous agents,” in *Intelligent Agents III Agent Theories, Architectures, and Languages*, Springer, Berlin, Heidelberg, 1997, pp. 21–35. doi: 10.1007/BFb0013570.

[31] M. Wooldridge and N. R. Jennings, “Intelligent agents: theory and practice,” *Knowl Eng Rev*, vol. 10, no. 02, pp. 115–152, Jun. 1995, doi: 10.1017/S0269888900008122.

[32] N. R. Jennings, K. Sycara, and M. Wooldridge, “A Roadmap of Agent Research and Development,” *Auton Agent Multi Agent Syst*, vol. 1, no. 1, pp. 7–38, 1998, doi: 10.1023/A:101009040.

[33] P. Leitão and S. Karnouskos, *Industrial Agents : Emerging Applications of Software Agents in Industry*. Elsevier Science and Technology Books, Inc., 2015.

[34] G. Peregud, J. Zubek, M. Ganzha, and M. Paprzycki, “Implementing an eXAT-based distributed monitoring system prototype,” *Computer Science and Information Systems*, vol. 9, no. 3, pp. 1249–1285, 2012, doi: 10.2298/CSIS120108028P.

[35] M. Piotrowski and W. Turek, “Software agents mobility using process migration mechanism in distributed Erlang,” in *Proceedings of the twelfth ACM SIGPLAN workshop on Erlang - Erlang ’13*, New York, New York, USA: ACM Press, 2013, p. 43. doi: 10.1145/2505305.2505312.

[36] A. Di Stefano, F. Gangemi, and C. Santoro, “ERESYE: Artificial Intelligence in Erlang Programs,” in *Proceedings of the 2005 ACM SIGPLAN workshop on Erlang - ERLANG ’05*, New York, New York, USA: ACM Press, 2005, pp. 62–71. doi: 10.1145/1088361.1088373.

[37] T. I. M. Berners-lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Sci Am*, pp. 35–43, 2001, Accessed: Nov. 18, 2023. [Online]. Available: https://redirect.cs.umbc.edu/courses/graduate/691/spring13/01/papers/semanticWebSciAm.pdf

[38] S. Grimm, A. Abecker, J. Völker, and R. Studer, “Ontologies and the Semantic Web,” in *Handbook of Semantic Web Technologies*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 507–579. doi: 10.1007/978-3-540-92913-0_13.

[39] Z. Pan, T. Zhu, H. Liu, and H. Ning, “A survey of RDF management technologies and benchmark datasets,” *J Ambient Intell Humaniz Comput*, vol. 9, no. 5, pp. 1693–1704, Oct. 2018, doi: 10.1007/s12652-018-0876-2.

[40] J. Hendler, “Agents and the Semantic Web,” *IEEE Intell Syst*, vol. 16, no. 2, pp. 30–37, Mar. 2001, doi: 10.1109/5254.920597.

[41] N. Shadbolt, T. Berners-Lee, and W. Hall, “The Semantic Web Revisited,” *IEEE Intell Syst*, vol. 21, no. 3, pp. 96–101, May 2006, doi: 10.1109/MIS.2006.62.

[42] G. Demir, P. Chatterjee, and D. Pamucar, “Sensitivity analysis in multi-criteria decision making: A state-of-the-art research perspective using bibliometric analysis,” *Expert Syst Appl*, vol. 237, p. 121660, Mar. 2024, doi: 10.1016/j.eswa.2023.121660.

[43] E. Mulliner, N. Malys, and V. Maliene, “Comparative analysis of MCDM methods for the assessment of sustainable housing affordability,” *Omega (Westport)*, vol. 59, pp. 146–156, Mar. 2016, doi: 10.1016/j.omega.2015.05.013.

[44] H. Taherdoost and M. Madanchian, “Multi-Criteria Decision Making (MCDM) Methods and Concepts,” *Encyclopedia*, vol. 3, no. 1, pp. 77–87, Jan. 2023, doi: 10.3390/encyclopedia3010006.

[45] M. M. D. Widianta, T. Rizaldi, D. P. S. Setyohadi, and H. Y. Riskiawan, “Comparison of Multi-Criteria Decision Support Methods (AHP, TOPSIS, SAW & PROMENTHEE)

for Employee Placement,” *J Phys Conf Ser*, vol. 953, no. 1, p. 012116, Jan. 2018, doi: 10.1088/1742-6596/953/1/012116.

[46] A. Afshari, M. Mojahed, and R. M. Yusuff, “Simple Additive Weighting approach to Personnel Selection problem,” *International Journal of Innovation, Management and Technology*, vol. 1, no. 5, pp. 511–515, 2010, doi: 10.7763/IJIMT.2010.V1.89.

[47] S. Gagnon, “Rebranding IS/IT Management Programs: The Case of Business Technology Management (BTM) in Canada,” in *EDSIGCON Proceedings*, Clearwater, FL: ISCAP, Nov. 2022, pp. 1–14. Accessed: Dec. 08, 2023. [Online]. Available: https://iscap.us/proceedings/2022/pdf/5850.pdf

[48] S. Gagnon, “Business Technology Management as Transdisciplinary IS-IT Competency Framework,” in *ICIS 2020 Proceedings. 8*, AIS, Dec. 2020, pp. 1–9. Accessed: Dec. 09, 2023. [Online]. Available: https://aisel.aisnet.org/icis2020/digital_learning_env/digital_learning_env/8

[49] A. DEMİRCİ, “Multi-Criteria Decision-Making Technique for Personnel Selection: PSI Sample,” *Toros Üniversitesi İİSBF Sosyal Bilimler Dergisi*, vol. 9, no. Special Issue 2nd International Symposium of Sustainable Logistics “Circular Economy,” pp. 10–17, Oct. 2022, doi: 10.54709/iisbf.1167228.

[50] J. R. Hackman and G. R. Oldham, “Motivation through the design of work: test of a theory,” *Organ Behav Hum Perform*, vol. 16, no. 2, pp. 250–279, Aug. 1976, doi: 10.1016/0030-5073(76)90016-7.

[51] P. K. D. Pramanik, S. Biswas, S. Pal, D. Marinković, and P. Choudhury, “A Comparative Analysis of Multi-Criteria Decision-Making Methods for Resource Selection in Mobile Crowd Computing,” *Symmetry (Basel)*, vol. 13, no. 9, p. 1713, Sep. 2021, doi: 10.3390/sym13091713.

[52] E. Cakir and M. Doğaner, “Application of Multi-Criteria Decision Making Approaches for Personnel Selection in A Private Hospital,” in *Recent Researches On Social Sciences*, 1st ed., R. A. H. M. R. Dorczak, Ed., Kraków: Jagiellonian University, Institute of Public Affairs, 2018, pp. 247–258. Accessed: Nov. 08, 2023. [Online]. Available: https://ruj.uj.edu.pl/xmlui/bitstream/handle/item/66318/dorczak_arslan_musialik_recent_researches_on_social_sciences_2018.pdf?sequence=1&isAllowed=y#page=248

[53] A. Reza Afshari, R. Mohd Yusuff, T. Sai Hong, and Y. Bin Ismail, “A review of the applications of multi criteria decision making for personnel selection problem,” *African Journal of Business Management*, vol. 5, no. 28, 2011, Accessed: Nov. 03, 2023. [Online]. Available:

https://www.researchgate.net/publication/257289960_A_review_of_the_applications_of_multi_criteria_decision_making_for_personnel_selection_problem

[54] A. Agarwal, "Automating Decision-Making for Hiring Brilliant People While Taking Risk Factors Into Account: A Data Mining Approach," in *2023 International Conference on Disruptive Technologies (ICDT)*, IEEE, May 2023, pp. 293–296. doi: 10.1109/ICDT57929.2023.10150667.

[55] S. Guo, F. Alamudun, and T. Hammond, "RésuMatcher: A personalized résumé-job matching system," *Expert Syst Appl*, vol. 60, pp. 169–182, Oct. 2016, doi: 10.1016/j.eswa.2016.04.013.

[56] I. Ali, N. Mughal, Z. H. Khan, J. Ahmed, and G. Mujtaba, "Resume Classification System using Natural Language Processing and Machine Learning Techniques," *Mehran University Research Journal of Engineering and Technology*, vol. 41, no. 1, pp. 65–79, Jan. 2022, doi: 10.22581/muet1982.2201.07.

[57] C. Hewitt, "Planner: a language for proving theorems in robots," in *Proceedings of the 1st international joint conference on Artificial intelligence, IJCAI'69*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1969, pp. 295–301. [Online]. Available: https://www.ijcai.org/Proceedings/69/Papers/030.pdf

[58] C. Hewitt, "PLANNER: A Language for Manipulating Models and Proving Theorems in a Robot," 1970. [Online]. Available: https://dspace.mit.edu/bitstream/handle/1721.1/6171/AIM-168.pdf?sequence=2

[59] C. Hewitt, P. Bishop, and R. Steiger, "A universal modular ACTOR formalism for artificial intelligence," in *Proceedings of the 3rd international joint conference on Artificial intelligence*, Morgan Kaufmann Publishers Inc., 1973, pp. 235–245.

[60] G. A. Agha, I. A. Mason, S. F. Smith, and C. L. Talcott, "A foundation for actor computation," *Journal of Functional Programming*, vol. 7, no. 1, pp. 1–72, Jan. 1997, doi: 10.1017/S095679689700261X.

[61] G. Agha, "Concurrent object-oriented programming," *Commun ACM*, vol. 33, no. 9, pp. 125–141, Sep. 1990, doi: 10.1145/83880.84528.

[62] G. Agha, *Actors: A Model of Concurrent Computation in Distributed Systems*. MIT Press, 1986.

[63] G. Agha, S. Frolund, W. Y. Kim, R. Panwar, A. Patterson, and D. Sturman, "Abstraction and modularity mechanisms for concurrent computing," *IEEE Parallel & Distributed Technology: Systems & Applications*, vol. 1, no. 2, pp. 3–14, May 1993, doi: 10.1109/88.218170.

[64] G. Agha and C. Hewitt, “Concurrent programming using actors: Exploiting large-scale parallelism,” *In: Maheshwari S.N. (eds) Foundations of Software Technology and Theoretical Computer Science. FSTTCS 1985. Lecture Notes in Computer Science*, vol. 206, pp. 19–41, 1985, doi: 10.1007/3-540-16042-6_2.

[65] A. Yonezawa, J.-P. Briot, and E. Shibayama, “Object-oriented concurrent programming ABCL/1,” *ACM SIGPLAN Notices*, vol. 21, no. 11, pp. 258–268, Nov. 1986, doi: 10.1145/960112.28722.

[66] P. Haller and M. Odersky, “Scala Actors: Unifying thread-based and event-based programming,” *Theor Comput Sci*, vol. 410, no. 2–3, pp. 202–220, Feb. 2009, doi: 10.1016/j.tcs.2008.09.019.

[67] J. De Koster, T. Van Cutsem, and W. De Meuter, “43 years of actors: a taxonomy of actor models and their key properties,” in *Proceedings of the 6th International Workshop on Programming Based on Actors, Agents, and Decentralized Control - AGERE 2016*, New York, New York, USA: ACM Press, 2016, pp. 31–40. doi: 10.1145/3001886.3001890.

[68] A. Santi, “Engineering Agent-Oriented Technologies and Programming Languages for Computer Programming and Software Development,” PhD thesis, PhD thesis, Department of Electronics, Computer Science and Systems, Università di Bologna, 2013. [Online]. Available: http://amsdottorato.unibo.it/5634/1/santi_andrea_tesi.pdf

[69] J. Armstrong, “Making reliable distributed systems in the presence of software errors,” PhD Thesis, PhD Thesis,The Royal Institute of Technology Stockholm, Sweden, 2003. [Online]. Available: http://erlang.org/download/armstrong_thesis_2003.pdf

[70] J. Wang, R. Zhou, J. Li, and G. Wang, “A Distributed Rule Engine Based on Message-Passing Model to Deal with Big Data,” *Lecture Notes on Software Engineering*, vol. 2, no. 3, pp. 275–281, 2014, doi: 10.7763/LNSE.2014.V2.136.

[71] Shiow-Yang Wu, D. P. Miranker, and J. C. Browne, “Decomposition abstraction in parallel rule languages,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 7, no. 11, pp. 1164–1184, 1996, doi: 10.1109/71.544357.

[72] C. L. Forgy, “Rete: A fast algorithm for the many pattern/many object pattern match problem,” *Artif Intell*, vol. 19, no. 1, pp. 17–37, Sep. 1982, doi: 10.1016/0004-3702(82)90020-0.

[73] T. DONG, J. SHI, J. FAN, and L. ZHANG, “An Improved Rete Algorithm Based on Double Hash Filter and Node Indexing for Distributed Rule Engine,” *IEICE Trans Inf Syst*, vol. E96.D, no. 12, pp. 2635–2644, 2013, doi: 10.1587/transinf.E96.D.2635.

[74] K. U. Schmidt, R. Stuhmer, and L. Stojanovic, “Blending complex event processing with the RETE algorithm,” in *Proceedings of the 1st iCEP08 Workshop on Complex Event Processing for the Future Internet, CEUR Workshop Proceedings*, CEUR-WS, 2008. [Online]. Available: http://ceur-ws.org/Vol-412/

[75] R. B. Doorenbos, “Production Matching for Large Learning Systems (PhD thesis),” Ph.D. Thesis, Computer Science Department, Carnegie Mellon University, 1995. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a293105.pdf

[76] A. Boaye Belle, T. C. Lethbridge, M. Garzón, and O. O. Adesina, “Design and implementation of distributed expert systems: On a control strategy to manage the execution flow of rule activation,” *Expert Syst Appl*, vol. 96, pp. 129–148, Apr. 2018, doi: 10.1016/j.eswa.2017.11.033.

[77] C. Santoro, “An erlang framework for autonomous mobile robots,” in *Proceedings of the 2007 SIGPLAN workshop on Erlang Workshop - Erlang ’07*, New York, New York, USA: ACM Press, 2007, p. 85. doi: 10.1145/1292520.1292533.

[78] R. Zhou, G. Wang, J. Wang, and J. Li, “RUNES II: A Distributed Rule Engine Based on Rete Network in Cloud Computing,” *International Journal of Grid and Distributed Computing*, vol. 7, no. 6, pp. 91–110, Dec. 2014, doi: 10.14257/ijgdc.2014.7.6.08.

[79] S. I. Shapovalova, “Formalization of the Rules of Inference in Erlang,” *Mathematical and computer modelling. Series: Technical sciences*, no. 21, pp. 125–139, Nov. 2020, doi: 10.32626/2308-5916.2020-21.125-139.

[80] D. Byun and E. Suh, “Human resource management expert systems technology,” *Expert Syst*, vol. 11, no. 2, pp. 109–119, May 1994, doi: 10.1111/j.1468-0394.1994.tb00004.x.

[81] R. Storey Hooper, T. P. Galvin, R. A. Kilmer, and J. Liebowitz, “Use of an expert system in a personnel selection process,” *Expert Syst Appl*, vol. 14, no. 4, pp. 425–432, Apr. 1998, doi: 10.1016/S0957-4174(98)00002-5.

[82] M. S. Mehrabad and F. M. Brojeny, “The development of an expert system for effective selection and appointment of the jobs applicants in human resource management,” *Comput Ind Eng*, vol. 53, no. 2, pp. 306–312, Sep. 2007, doi: 10.1016/j.cie.2007.06.023.

[83] C.-F. Chien and L.-F. Chen, “Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry,” *Expert Syst Appl*, vol. 34, no. 1, pp. 280–290, Jan. 2008, doi: 10.1016/j.eswa.2006.09.003.

[84] E. Akhlaghi, “A Rough-set Based Approach to Design an Expert System for Personnel Selection,” *World Academy of Science, Engineering and Technology, International*

Journal of Industrial and Manufacturing Engineering, vol. 5, no. 6, pp. 999–1002, Mar. 2011, Accessed: Nov. 13, 2023. [Online]. Available: https://zenodo.org/records/1082331

[85] E. Jereb, U. Rajkovic, and V. Rajkovic, "A Hierarchical Multi-Attribute System Approach to Personnel Selection," *International Journal of Selection and Assessment*, vol. 13, no. 3, pp. 198–205, Sep. 2005, doi: 10.1111/j.1468-2389.2005.00315.x.

[86] M. L. Gargano, R. A. Marose, and L. von Kleeck, "An application of artificial neural networks and genetic algorithms to personnel selection in the financial industry," in *Proceedings First International Conference on Artificial Intelligence Applications on Wall Street*, IEEE Comput. Soc. Press, 1991, pp. 257–262. doi: 10.1109/AIAWS.1991.236592.

[87] V. Cho and E. W. T. Ngai, "Data mining for selection of insurance sales agents," *Expert Syst*, vol. 20, no. 3, pp. 123–132, Jul. 2003, doi: 10.1111/1468-0394.00235.

[88] A. Drigas, S. Kouremenos, S. Vrettos, J. Vrettaros, and D. Kouremenos, "An expert system for job matching of the unemployed," *Expert Syst Appl*, vol. 26, no. 2, pp. 217–224, Feb. 2004, doi: 10.1016/S0957-4174(03)00136-2.

[89] Liang-Chih Huang, Kuo-Shu Huang, Hsiu-Ping Huang, and Bih-Shiaw Jaw, "Applying fuzzy neural network in human resource selection system," in *IEEE Annual Meeting of the Fuzzy Information, 2004. Processing NAFIPS '04.*, IEEE, 2004, pp. 169-174 Vol.1. doi: 10.1109/NAFIPS.2004.1336271.

[90] A. Rashidi, F. Jazebi, and I. Brilakis, "Neurofuzzy Genetic System for Selection of Construction Project Managers," *J Constr Eng Manag*, vol. 137, no. 1, pp. 17–29, Jan. 2011, doi: 10.1061/(ASCE)CO.1943-7862.0000200.

[91] F. Jazebi and A. Rashidi, "AN AUTOMATED PROCEDURE FOR SELECTING PROJECT MANAGERS IN CONSTRUCTION FIRMS," *Journal of Civil Engineering and Management*, vol. 19, no. 1, pp. 97–106, Jan. 2013, doi: 10.3846/13923730.2012.738707.

[92] T. Berners-Lee, "Information Management: A Proposal," *No. CERN-DD-89-001-OC.*, 1989, [Online]. Available: https://cds.cern.ch/record/369245/files/dd-89-001.pdf

[93] L. M. O. Machado, R. R. Souza, and M. da G. Simões, "Semantic Web or Web of Data? A diachronic study (1999 to 2017) of the publications of Tim Berners-Lee and the World Wide Web Consortium," 2018.

[94] T. Berners-Lee, "Semantic Web Development: Technical Proposal," W3C. Accessed: Nov. 22, 2023. [Online]. Available: https://www.w3.org/2000/01/sw/DevelopmentProposal

[95] S. Runde and A. Fay, “Software Support for Building Automation Requirements Engineering—An Application of Semantic Web Technologies in Automation,” *IEEE Trans Industr Inform*, vol. 7, no. 4, pp. 723–730, Nov. 2011, doi: 10.1109/TII.2011.2166784.

[96] S. Runde, A. Fay, and W.-O. Wutzke, “Knowledge-based Requirement-Engineering of building automation systems by means of Semantic Web technologies,” in *2009 7th IEEE International Conference on Industrial Informatics*, IEEE, Jun. 2009, pp. 267–272. doi: 10.1109/INDIN.2009.5195815.

[97] A. Alinezhad and J. Khalili, *New Methods and Applications in Multiple Attribute Decision Making (MADM)*, vol. 277. in International Series in Operations Research & Management Science, vol. 277. Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-030-15009-9.

[98] A. Valipour, H. Sarvari, and J. Tamošaitiene, “Risk Assessment in PPP Projects by Applying Different MCDM Methods and Comparative Results Analysis,” *Adm Sci*, vol. 8, no. 4, p. 80, Dec. 2018, doi: 10.3390/admsci8040080.

[99] C. Srisawat and J. Payakpate, “COMPARISON OF MCDM METHODS FOR INTERCROP SELECTION IN RUBBER PLANTATIONS,” *Journal of ICT*, vol. 15, no. 1, pp. 165–182, 2016, doi: 10.32890/jict.

[100] C.-L. Hwang, Y.-J. Lai, and T.-Y. Liu, “A new approach for multiple objective decision making,” *Comput Oper Res*, vol. 20, no. 8, pp. 889–899, Oct. 1993, doi: 10.1016/0305-0548(93)90109-V.

[101] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making Methods and Applications A State-of-the-Art Survey*, vol. 186. in Lecture Notes in Economics and Mathematical Systems, vol. 186. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981. doi: 10.1007/978-3-642-48318-9.

[102] M. Keshavarz Ghorabaee, E. K. Zavadskas, L. Olfat, and Z. Turskis, “Multi-Criteria Inventory Classification Using a New Method of Evaluation Based on Distance from Average Solution (EDAS),” *Informatica*, vol. 26, no. 3, pp. 435–451, Jan. 2015, doi: 10.15388/Informatica.2015.57.

[103] A. Alinezhad and J. Khalili, “MABAC Method,” 2019, pp. 193–198. doi: 10.1007/978-3-030-15009-9_25.

[104] D. Pamučar and G. Ćirović, “The selection of transport and handling resources in logistics centers using Multi-Attributive Border Approximation area Comparison (MABAC),”

Expert Syst Appl, vol. 42, no. 6, pp. 3016–3028, Apr. 2015, doi: 10.1016/j.eswa.2014.11.057.

[105] E. K. Zavadskas and Z. Turskis, "A NEW ADDITIVE RATIO ASSESSMENT (ARAS) METHOD IN MULTICRITERIA DECISION-MAKING / NAUJAS ADITYVINIS KRITERIJŲ SANTYKIŲ ĮVERTINIMO METODAS (ARAS) DAUGIAKRITERINIAMS UŽDAVINIAMS SPRĘSTI," *Technological and Economic Development of Economy*, vol. 16, no. 2, pp. 159–172, Jun. 2010, doi: 10.3846/tede.2010.10.

[106] A. Alinezhad and J. Khalili, "ARAS Method," in *International Series in Operations Research and Management Science*, vol. 277, Springer New York LLC, 2019, pp. 67–71. doi: 10.1007/978-3-030-15009-9_9.

[107] K. R. Maccrimmon, "Decisionmaking among multiple-attribute alternatives: a survey and consolidated approach," Dec. 1968. Accessed: Nov. 01, 2023. [Online]. Available: https://apps.dtic.mil/sti/citations/tr/AD0681005

[108] E. Kazimieras Zavadskas and V. Šarka, "The new method of multicriteria complex proportional assessment of projects," *Technol. Econ. Dev. Econ*, vol. 1, no. 3, pp. 131–139, 1994, [Online]. Available: https://www.researchgate.net/publication/285902199

[109] A. Alinezhad and J. Khalili, "COPRAS Method," in *International Series in Operations Research and Management Science*, vol. 277, Springer New York LLC, 2019, pp. 87–91. doi: 10.1007/978-3-030-15009-9_12.

[110] L. Duckstein and S. Opricovic, "Multiobjective optimization in river basin development," *Water Resour Res*, vol. 16, no. 1, pp. 14–20, Feb. 1980, doi: 10.1029/WR016i001p00014.

[111] S. Opricovic and G.-H. Tzeng, "Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS," *Eur J Oper Res*, vol. 156, no. 2, pp. 445–455, Jul. 2004, doi: 10.1016/S0377-2217(03)00020-1.

[112] M. Yazdani, P. Zarate, E. Kazimieras Zavadskas, and Z. Turskis, "A combined compromise solution (CoCoSo) method for multi-criteria decision-making problems," *Management Decision*, vol. 57, no. 9, pp. 2501–2519, Oct. 2019, doi: 10.1108/MD-05-2017-0458.

[113] Ž. Stević, D. Pamučar, A. Puška, and P. Chatterjee, "Sustainable supplier selection in healthcare industries using a new MCDM method: Measurement of alternatives and ranking according to COmpromise solution (MARCOS)," *Comput Ind Eng*, vol. 140, p. 106231, Feb. 2020, doi: 10.1016/j.cie.2019.106231.

[114] M. Žižović, D. Pamučar, M. Albijanić, P. Chatterjee, and I. Pribićević, “Eliminating Rank Reversal Problem Using a New Multi-Attribute Model—The RAFSI Method,” *Mathematics*, vol. 8, no. 6, p. 1015, Jun. 2020, doi: 10.3390/math8061015.

[115] R. W. Saaty, “The analytic hierarchy process—what it is and how it is used,” *Mathematical Modelling*, vol. 9, no. 3–5, pp. 161–176, 1987, doi: 10.1016/0270-0255(87)90473-8.

[116] A. Charnes, W. W. Cooper, and E. Rhodes, “Measuring the efficiency of decision making units,” *Eur J Oper Res*, vol. 2, no. 6, pp. 429–444, Nov. 1978, doi: 10.1016/0377-2217(78)90138-8.

[117] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: 10.1016/S0019-9958(65)90241-X.

[118] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications-A State-of-the-Art Survey*, vol. 186. in Lecture Notes in Economics and Mathematical Systems, vol. 186. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981. doi: 10.1007/978-3-642-48318-9.

[119] A. Charnes and W. W. Cooper, “Management Models and Industrial Applications of Linear Programming,” *Manage Sci*, vol. 4, no. 1, pp. 38–91, Oct. 1957, doi: 10.1287/mnsc.4.1.38.

[120] J. L. Kolodner, “Understanding creativity: A case-based approach,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 837 LNAI, Springer Verlag, 1994, pp. 1–20. doi: 10.1007/3-540-58330-0_73.

[121] D. Ju-Long, “Control problems of grey systems,” *Syst Control Lett*, vol. 1, no. 5, pp. 288–294, Mar. 1982, doi: 10.1016/S0167-6911(82)80025-X.

[122] T. L. Saaty, *Decision making with dependence and feedback : the analytic network process : the organization and prioritization of complexity*. RWS Publications, 1996.

[123] P. J. M. van Laarhoven and W. Pedrycz, “A fuzzy extension of Saaty’s priority theory,” *Fuzzy Sets Syst*, vol. 11, no. 1–3, pp. 229–241, 1983, doi: 10.1016/S0165-0114(83)80082-7.

[124] R. Banayoun, B. Roy, and N. Sussman, “Manual de Reference du Programme Electre, Note de Synthese et Formation 25,” *Direction Scientifique SEMA*, 1966.

[125] J. P. Brans, Ph. Vincke, and B. Mareschal, “How to select and how to rank projects: The Promethee method,” *Eur J Oper Res*, vol. 24, no. 2, pp. 228–238, Feb. 1986, doi: 10.1016/0377-2217(86)90044-5.

[126] S. Opricovic, “Multicriteria Optimization of Civil Engineering Systems,” PhD Thesis, University of Belgrade, Belgrade, Serbia, 1998.

[127] C.-T. Chen, “Extensions of the TOPSIS for group decision-making under fuzzy environment,” *Fuzzy Sets Syst*, vol. 114, no. 1, pp. 1–9, Aug. 2000, doi: 10.1016/S0165-0114(97)00377-1.

[128] Y.-J. Lai, T.-Y. Liu, and C.-L. Hwang, “TOPSIS for MODM,” *Eur J Oper Res*, vol. 76, no. 3, pp. 486–500, Aug. 1994, doi: 10.1016/0377-2217(94)90282-8.

[129] A. Gabus and E. Fontela, “World Problems, An Invitation to Further Thought within the Framework of DEMATEL,” *Battelle Geneva Research Center*, pp. 1–8, 1972.

[130] B. Matarazzo, “Preference ranking global frequencies in multicriterion analysis (Pragma),” *Eur J Oper Res*, vol. 36, no. 1, pp. 36–49, Jul. 1988, doi: 10.1016/0377-2217(88)90005-7.

[131] C. W. Churchman and R. L. Ackoff, “An Approximate Measure of Value,” *Journal of the Operations Research Society of America*, vol. 2, no. 2, pp. 172–187, May 1954, doi: 10.1287/opre.2.2.172.

[132] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives*. Cambridge University Press, 1993. doi: 10.1017/CBO9781139174084.

[133] J. Rezaei, “Best-worst multi-criteria decision-making method,” *Omega (Westport)*, vol. 53, pp. 49–57, Jun. 2015, doi: 10.1016/j.omega.2014.11.009.

[134] D. L. Olson, “Smart,” in *Decision Aids for Selection Problems*, New York, NY: Springer New York, 1996, pp. 34–48. doi: 10.1007/978-1-4612-3982-6_4.

[135] L. Mikhailov and M. G. Singh, “Fuzzy analytic network process and its application to the development of decision support systems,” *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 33, no. 1, pp. 33–41, Feb. 2003, doi: 10.1109/TSMCC.2003.809354.

[136] S. H. Zanakis, A. Solomon, N. Wishart, and S. Dublish, “Multi-attribute decision making: A simulation comparison of select methods,” *Eur J Oper Res*, vol. 107, no. 3, pp. 507–529, Jun. 1998, doi: 10.1016/S0377-2217(97)00147-1.

[137] E. Triantaphyllou and C.-T. Lin, “Development and evaluation of five fuzzy multiattribute decision-making methods,” *International Journal of Approximate Reasoning*, vol. 14, no. 4, pp. 281–310, May 1996, doi: 10.1016/0888-613X(95)00119-2.

[138] J. Ruby Annette, A. Banu, and P. Subash Chandran, “Comparison of Multi Criteria Decision Making Algorithms for Ranking Cloud Renderfarm Services,” *Indian J Sci Technol*, vol. 9, no. 31, Aug. 2016, doi: 10.17485/ijst/2016/v9i31/93467.

[139] M. Dağdeviren, “A hybrid multi-criteria decision-making model for personnel selection in manufacturing systems,” *J Intell Manuf*, vol. 21, no. 4, pp. 451–460, Aug. 2010, doi: 10.1007/s10845-008-0200-7.

[140] F. Rajabi, H. Molaeifar, M. Jahangiri, S. Taheri, S. Banaee, and P. Farhadi, “Occupational stressors among firefighters: application of multi-criteria decision making (MCDM)Techniques,” *Heliyon*, vol. 6, no. 4, p. e03820, Apr. 2020, doi: 10.1016/j.heliyon.2020.e03820.

[141] S. Alp and T. K. Özkan, “Job Choice with Multi-Criteria Decision Making Approach in a Fuzzy Environment,” *International Review of Management and Marketing* |, vol. 5, no. 3, pp. 165–172, 2015, Accessed: Oct. 31, 2023. [Online]. Available: https://dergipark.org.tr/en/pub/irmm/issue/32085/355088

[142] M. Pekkaya, “Career Preference of University Students: An Application of MCDM Methods,” *Procedia Economics and Finance*, vol. 23, pp. 249–255, 2015, doi: 10.1016/S2212-5671(15)00486-4.

[143] A. Purwita and A. Subriadi, “Literature Review – Using Multi-Criteria Decision-Making Methods in Information Technology (IT) Investment,” in *Proceedings of the Proceedings of the 1st International Conference on Business, Law And Pedagogy, ICBLP 2019, 13-15 February 2019, Sidoarjo, Indonesia*, EAI, Feb. 2019. doi: 10.4108/eai.13-2-2019.2286076.

[144] A. I. Marqués, V. García, and J. S. Sánchez, “Ranking-based MCDM models in financial management applications: analysis and emerging challenges,” *Progress in Artificial Intelligence*, vol. 9, no. 3, pp. 171–193, Sep. 2020, doi: 10.1007/s13748-020-00207-1.

[145] D. Danesh, M. J. Ryan, and A. Abbasi, “Multi-criteria Decision-making Methods for Project Portfolio Management: A Literature Review,” *International Journal of Management and Decision Making*, vol. 16, no. 1, p. 1, 2017, doi: 10.1504/IJMDM.2017.10006139.

[146] K. Alkaradaghi, S. S. Ali, N. Al-Ansari, J. Laue, and A. Chabuk, “Landfill Site Selection Using MCDM Methods and GIS in the Sulaimaniyah Governorate, Iraq,” *Sustainability*, vol. 11, no. 17, p. 4530, Aug. 2019, doi: 10.3390/su11174530.

[147] B. Özkan, E. Özceylan, and İ. Sarıçiçek, “GIS-based MCDM modeling for landfill site suitability analysis: A comprehensive review of the literature,” *Environmental Science and Pollution Research*, vol. 26, no. 30, pp. 30711–30730, Oct. 2019, doi: 10.1007/s11356-019-06298-1.

[148] A. Coban, I. F. Ertis, and N. A. Cavdaroglu, “Municipal solid waste management via multi-criteria decision making methods: A case study in Istanbul, Turkey,” *J Clean Prod*, vol. 180, pp. 159–167, Apr. 2018, doi: 10.1016/j.jclepro.2018.01.130.

[149] E. Arıkan, Z. T. Şimşit-Kalender, and Ö. Vayvay, “Solid waste disposal methodology selection using multi-criteria decision making methods and an application in Turkey,” *J Clean Prod*, vol. 142, pp. 403–412, Jan. 2017, doi: 10.1016/j.jclepro.2015.10.054.

[150] I. Emovon and O. S. Oghenenyerovwho, “Application of MCDM method in material selection for optimal design: A review,” *Results in Materials*, vol. 7, p. 100115, Sep. 2020, doi: 10.1016/j.rinma.2020.100115.

[151] M. Stojčić, E. Zavadskas, D. Pamučar, Ž. Stević, and A. Mardani, “Application of MCDM Methods in Sustainability Engineering: A Literature Review 2008–2018,” *Symmetry (Basel)*, vol. 11, no. 3, p. 350, Mar. 2019, doi: 10.3390/sym11030350.

[152] B. Singaravel, D. P. Shankar, and L. Prasanna, “Application of MCDM Method for the Selection of Optimum Process Parameters in Turning Process,” *Mater Today Proc*, vol. 5, no. 5, pp. 13464–13471, 2018, doi: 10.1016/j.matpr.2018.02.341.

[153] M. Shafiee, “Maintenance strategy selection problem: an MCDM overview,” *J Qual Maint Eng*, vol. 21, no. 4, pp. 378–402, Oct. 2015, doi: 10.1108/JQME-09-2013-0063.

[154] P. Chowdhury and S. K. Paul, “Applications of MCDM methods in research on corporate sustainability,” *Management of Environmental Quality: An International Journal*, vol. 31, no. 2, pp. 385–405, Feb. 2020, doi: 10.1108/MEQ-12-2019-0284.

[155] M. Brzozowski and I. Birfer, “Applications of MCDM Methods in the ERP System Selection Process in Enterprises,” *Handel Wewnętrzny*, vol. 3, no. 368, pp. 40–52, 2017.

[156] E. Kazimieras Zavadskas, J. Antucheviciene, and P. Chatterjee, “Multiple-Criteria Decision-Making (MCDM) Techniques for Business Processes Information Management,” *Information*, vol. 10, no. 1, p. 4, Dec. 2018, doi: 10.3390/info10010004.

[157] E. Mukul, G. Büyüközkan, and M. Güler, “Evaluation of Digital Marketing Technologies with Mcdm Methods,” in *6th International Conference on New Ideas in Management Economics and Accounting*, Paris, 2019, pp. 19–21. Accessed: Oct. 31, 2023. [Online]. Available: https://www.dpublication.com/wp-content/uploads/2019/04/7-IMEA-O-230.pdf

[158] A. Jusoh, A. Mardani, R. Omar, D. Štreimikienė, Z. Khalifah, and A. Sharifara, “APPLICATION OF MCDM APPROACH TO EVALUATE THE CRITICAL SUCCESS FACTORS OF TOTAL QUALITY MANAGEMENT IN THE HOSPITALITY

INDUSTRY," *Journal of Business Economics and Management*, vol. 19, no. 2, pp. 399–416, Oct. 2018, doi: 10.3846/jbem.2018.5538.

[159] A. T. de Almeida, M. H. Alencar, T. V. Garcez, and R. J. P. Ferreira, "A systematic literature review of multicriteria and multi-objective models applied in risk management," *IMA Journal of Management Mathematics*, vol. 28, no. 2, pp. 153–184, Apr. 2017, doi: 10.1093/imaman/dpw021.

[160] K. Govindan, S. Rajendran, J. Sarkis, and P. Murugesan, "Multi criteria decision making approaches for green supplier evaluation and selection: a literature review," *J Clean Prod*, vol. 98, pp. 66–83, Jul. 2015, doi: 10.1016/j.jclepro.2013.06.046.

[161] A. Yildiz and A. Y. Yayla, "MULTI-CRITERIA DECISION-MAKING METHODS FOR SUPPLIER SELECTION: A LITERATURE REVIEW," *The South African Journal of Industrial Engineering*, vol. 26, no. 2, p. 158, Aug. 2015, doi: 10.7166/26-2-1010.

[162] V. B. Schramm, L. P. B. Cabral, and F. Schramm, "Approaches for supporting sustainable supplier selection - A literature review," *J Clean Prod*, vol. 273, p. 123089, Nov. 2020, doi: 10.1016/j.jclepro.2020.123089.

[163] Valerio Salomon, *Multi-Criteria Methods and Techniques Applied to Supply Chain Management*. London: InTech, 2018. doi: 10.5772/intechopen.71296.

[164] M. Shao, Z. Han, J. Sun, C. Xiao, S. Zhang, and Y. Zhao, "A review of multi-criteria decision making applications for renewable energy site selection," *Renew Energy*, vol. 157, pp. 377–403, Sep. 2020, doi: 10.1016/j.renene.2020.04.137.

[165] A. Kumar *et al.*, "A review of multi criteria decision making (MCDM) towards sustainable renewable energy development," *Renewable and Sustainable Energy Reviews*, vol. 69, pp. 596–609, Mar. 2017, doi: 10.1016/j.rser.2016.11.191.

[166] İ. Kaya, M. Çolak, and F. Terzi, "Use of MCDM techniques for energy policy and decision-making problems: A review," *Int J Energy Res*, vol. 42, no. 7, pp. 2344–2372, Jun. 2018, doi: 10.1002/er.4016.

[167] I. Siksnelyte-Butkiene, E. K. Zavadskas, and D. Streimikiene, "Multi-Criteria Decision-Making (MCDM) for the Assessment of Renewable Energy Technologies in a Household: A Review," *Energies (Basel)*, vol. 13, no. 5, p. 1164, Mar. 2020, doi: 10.3390/en13051164.

[168] J. Antucheviciene, Z. Kala, M. Marzouk, and E. R. Vaidogas, "Decision Making Methods and Applications in Civil Engineering," *Math Probl Eng*, vol. 2015, pp. 1–3, 2015, doi: 10.1155/2015/160569.

[169] V. Penadés-Plà, T. García-Segura, J. Martí, and V. Yepes, “A Review of Multi-Criteria Decision-Making Methods Applied to the Sustainable Bridge Design,” *Sustainability*, vol. 8, no. 12, p. 1295, Dec. 2016, doi: 10.3390/su8121295.

[170] E. K. Zavadskas, J. Antuchevičienė, and O. Kaplinski, “MULTI-CRITERIA DECISION MAKING IN CIVIL ENGINEERING. PART II – APPLICATIONS,” *Engineering Structures and Technologies*, vol. 7, no. 4, pp. 151–167, Apr. 2016, doi: 10.3846/2029882X.2016.1139664.

[171] S. El Gibari, T. Gómez, and F. Ruiz, “Building composite indicators using multicriteria methods: a review,” *Journal of Business Economics*, vol. 89, no. 1, pp. 1–24, Feb. 2019, doi: 10.1007/s11573-018-0902-z.

[172] J. Si, L. Marjanovic-Halburd, F. Nasiri, and S. Bell, “Assessment of building-integrated green technologies: A review and case study on applications of Multi-Criteria Decision Making (MCDM) method,” *Sustain Cities Soc*, vol. 27, pp. 106–115, Nov. 2016, doi: 10.1016/j.scs.2016.06.013.

[173] T. Tan, G. Mills, E. Papadonikolaki, and Z. Liu, “Combining multi-criteria decision making (MCDM) methods with building information modelling (BIM): A review,” *Autom Constr*, vol. 121, p. 103451, Jan. 2021, doi: 10.1016/j.autcon.2020.103451.

[174] M. PAVLOVSKIS, J. ANTUCHEVICIENE, and D. MIGILINSKAS, “Application of MCDM and BIM for Evaluation of Asset Redevelopment Solutions,” *Studies in Informatics and Control*, vol. 25, no. 3, Sep. 2016, doi: 10.24846/v25i3y201603.

[175] G. Tran Thi Hoang, L. Dupont, and M. Camargo, “Application of Decision-Making Methods in Smart City Projects: A Systematic Literature Review,” *Smart Cities*, vol. 2, no. 3, pp. 433–452, Sep. 2019, doi: 10.3390/smartcities2030027.

[176] Ž. Morkūnaitė, D. Kalibatas, and D. Kalibatienė, “A BIBLIOMETRIC DATA ANALYSIS OF MULTI-CRITERIA DECISION MAKING METHODS IN HERITAGE BUILDINGS,” *JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT*, vol. 25, no. 2, pp. 76–99, Feb. 2019, doi: 10.3846/jcem.2019.8315.

[177] S. L. Gebre, D. Cattrysse, and J. Van Orshoven, “Multi-Criteria Decision-Making Methods to Address Water Allocation Problems: A Systematic Review,” *Water (Basel)*, vol. 13, no. 2, p. 125, Jan. 2021, doi: 10.3390/w13020125.

[178] M. Zare *et al.*, “Multi-criteria decision making approach in E-learning: A systematic review and classification,” *Appl Soft Comput*, vol. 45, pp. 108–128, Aug. 2016, doi: 10.1016/j.asoc.2016.04.020.

[179] S. Pal, P. Kanti Dutta Pramanik, M. Alsulami, A. Nayyar, M. Zarour, and P. Choudhury, "Using DEMATEL for Contextual Learner Modeling in Personalized and Ubiquitous Learning," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3981–4001, 2021, doi: 10.32604/cmc.2021.017966.

[180] N. Z. Khan, T. S. A. Ansari, A. N. Siddiquee, and Z. A. Khan, "Selection of E-learning websites using a novel Proximity Indexed Value (PIV) MCDM method," *Journal of Computers in Education*, vol. 6, no. 2, pp. 241–256, Jun. 2019, doi: 10.1007/s40692-019-00135-7.

[181] G. Yannis, A. Kopsacheili, A. Dragomanovits, and V. Petraki, "State-of-the-art review on multi-criteria decision-making in the transport sector," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 7, no. 4, pp. 413–431, Aug. 2020, doi: 10.1016/j.jtte.2020.05.005.

[182] M. Kiciński and K. Solecka, "Application of MCDA/MCDM methods for an integrated urban public transportation system – case study, city of Cracow," *Archives of Transport*, vol. 46, no. 2, pp. 71–84, Jun. 2018, doi: 10.5604/01.3001.0012.2107.

[183] J. Camargo Pérez, M. H. Carrillo, and J. R. Montoya-Torres, "Multi-criteria approaches for urban passenger transport systems: a literature review," *Ann Oper Res*, vol. 226, no. 1, pp. 69–87, Mar. 2015, doi: 10.1007/s10479-014-1681-8.

[184] M. Nassereddine and H. Eskandari, "An integrated MCDM approach to evaluate public transportation systems in Tehran," *Transp Res Part A Policy Pract*, vol. 106, pp. 427–439, Dec. 2017, doi: 10.1016/j.tra.2017.10.013.

[185] M. D. Oliveira, I. Mataloto, and P. Kanavos, "Multi-criteria decision analysis for health technology assessment: addressing methodological challenges to improve the state of the art," *The European Journal of Health Economics*, vol. 20, no. 6, pp. 891–918, Aug. 2019, doi: 10.1007/s10198-019-01052-3.

[186] M. Gul, "A review of occupational health and safety risk assessment approaches based on multi-criteria decision-making methods and their fuzzy versions," *Human and Ecological Risk Assessment: An International Journal*, vol. 24, no. 7, pp. 1723–1760, Oct. 2018, doi: 10.1080/10807039.2018.1424531.

[187] M. Mutlu, G. Tuzkaya, and B. Sennaroğlu, "MULTI-CRITERIA DECISION MAKING TECHNIQUES FOR HEALTHCARE SERVICE QUALITY EVALUATION: A LITERATURE REVIEW," *Sigma J Eng & Nat Sci*, vol. 35, no. 3, pp. 501–512, 2017.

[188] I. Karacan, H. Tozan, and M. Karatas, "Multi Criteria Decision Methods in Health Technology Assessment: A Brief Literature Review," *Eurasian Journal of Health*

Technology Assessment Corresponding Author Ilknur KARACAN, vol. 1, no. 1, pp. 12–19, 2016, [Online]. Available: https://www.researchgate.net/publication/330563963

[189] A. Piegat and W. Sałabun, "Comparative Analysis of MCDM Methods for Assessing the Severity of Chronic Liver Disease," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 9119, Springer Verlag, 2015, pp. 228–238. doi: 10.1007/978-3-319-19324-3_21.

[190] M. Mathew and S. Sahu, "Comparison of new multi-criteria decision making methods for material handling equipment selection," *Management Science Letters*, vol. 8, no. 3, pp. 139–150, 2018, doi: 10.5267/j.msl.2018.1.004.

[191] I. Ghosh and S. Biswas, "A Comparative Analysis of Multi-criteria Decision Models for ERP Package Selection for Improving Supply Chain Performance," *Asia-Pacific Journal of Management Research and Innovation*, vol. 12, no. 3–4, pp. 250–270, Sep. 2016, doi: 10.1177/2319510X16688988.

[192] A. Nesticò and P. Somma, "Comparative Analysis of Multi-Criteria Methods for the Enhancement of Historical Buildings," *Sustainability*, vol. 11, no. 17, p. 4526, Aug. 2019, doi: 10.3390/su11174526.

[193] M. Moradian, V. Modanloo, and S. Aghaiee, "Comparative analysis of multi criteria decision making techniques for material selection of brake booster valve body," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 6, no. 5, pp. 526–534, Oct. 2019, doi: 10.1016/j.jtte.2018.02.001.

[194] A. M. Ghaleb, H. Kaid, A. Alsamhan, S. H. Mian, and L. Hidri, "Assessment and Comparison of Various MCDM Approaches in the Selection of Manufacturing Process," *Advances in Materials Science and Engineering*, vol. 2020, pp. 1–16, Apr. 2020, doi: 10.1155/2020/4039253.

[195] B. Ceballos, M. T. Lamata, and D. A. Pelta, "A comparative analysis of multi-criteria decision-making methods," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 315–322, Nov. 2016, doi: 10.1007/s13748-016-0093-1.

[196] M. Vuković, S. Pivac, and Z. Babić, "Comparative analysis of stock selection using a hybrid MCDM approach and modern portfolio theory," *Croatian Review of Economic, Business and Social Statistics*, vol. 6, no. 2, pp. 58–68, Dec. 2020, doi: 10.2478/crebss-2020-0011.

[197] W. Sałabun and A. Piegat, "Comparative analysis of MCDM methods for the assessment of mortality in patients with acute coronary syndrome," *Artif Intell Rev*, vol. 48, no. 4, pp. 557–571, Dec. 2017, doi: 10.1007/s10462-016-9511-9.

[198] H.-C. Lee and C.-T. Chang, "Comparative analysis of MCDM methods for ranking renewable energy sources in Taiwan," *Renewable and Sustainable Energy Reviews*, vol. 92, pp. 883–896, Sep. 2018, doi: 10.1016/j.rser.2018.05.007.

[199] P. Karande, E. K. Zavadskas, and S. Chakraborty, "A study on the ranking performance of some MCDM methods for industrial robot selection problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 3, pp. 399–422, Jun. 2016, doi: 10.5267/j.ijiec.2016.1.001.

[200] E. Harirchian, K. Jadhav, K. Mohammad, S. E. Aghakouchaki Hosseini, and T. Lahmer, "A Comparative Study of MCDM Methods Integrated with Rapid Visual Seismic Vulnerability Assessment of Existing RC Structures," *Applied Sciences*, vol. 10, no. 18, p. 6411, Sep. 2020, doi: 10.3390/app10186411.

[201] J. Sidhu and S. Singh, "Design and Comparative Analysis of MCDM-based Multi-dimensional Trust Evaluation Schemes for Determining Trustworthiness of Cloud Service Providers," *J Grid Comput*, vol. 15, no. 2, pp. 197–218, Jun. 2017, doi: 10.1007/s10723-017-9396-0.

[202] S. A. A. Alrababah, K. H. Gan, and T.-P. Tan, "Comparative analysis of MCDM methods for product aspect ranking: TOPSIS and VIKOR," in *2017 8th International Conference on Information and Communication Systems (ICICS)*, IEEE, Apr. 2017, pp. 76–81. doi: 10.1109/IACS.2017.7921949.

[203] S. K. Kaya, "Evaluation of the Effect of COVID-19 on Countries' Sustainable Development Level: A comparative MCDM framework," *Operational Research in Engineering Sciences: Theory and Applications*, vol. 3, no. 3, p. 101, Dec. 2020, doi: 10.31181/oresta20303101k.

[204] T. Li, A. Li, and X. Guo, "The sustainable development-oriented development and utilization of renewable energy industry——A comprehensive analysis of MCDM methods," *Energy*, vol. 212, p. 118694, Dec. 2020, doi: 10.1016/j.energy.2020.118694.

[205] R. Sun, Z. Gong, G. Gao, and A. A. Shah, "Comparative analysis of Multi-Criteria Decision-Making methods for flood disaster risk in the Yangtze River Delta," *International Journal of Disaster Risk Reduction*, vol. 51, p. 101768, Dec. 2020, doi: 10.1016/j.ijdrr.2020.101768.

[206] P. Madhu, C. Sowmya Dhanalakshmi, and M. Mathew, "Multi-criteria decision-making in the selection of a suitable biomass material for maximum bio-oil yield during pyrolysis," *Fuel*, vol. 277, p. 118109, Oct. 2020, doi: 10.1016/j.fuel.2020.118109.

[207] S. Hezer, E. Gelmez, and E. Özceylan, “Comparative analysis of TOPSIS, VIKOR and COPRAS methods for the COVID-19 Regional Safety Assessment,” *J Infect Public Health*, vol. 14, no. 6, pp. 775–786, Jun. 2021, doi: 10.1016/j.jiph.2021.03.003.

[208] K.-G. MEHDI, A. MAGHSOUD, Z. EDMUNDAS KAZIMIERAS, T. ZENONAS, and A. JURGITA, “A Comparative Analysis of the Rank Reversal Phenomenon in the EDAS and TOPSIS Methods,” *Econ Comput Econ Cybern Stud Res*, vol. 52, no. 3/2018, pp. 121–134, Sep. 2018, doi: 10.24818/18423264/52.3.18.08.

[209] N. Kokaraki, C. J. Hopfe, E. Robinson, and E. Nikolaidou, “Testing the reliability of deterministic multi-criteria decision-making methods using building performance simulation,” *Renewable and Sustainable Energy Reviews*, vol. 112, pp. 991–1007, Sep. 2019, doi: 10.1016/j.rser.2019.06.018.

[210] S. Dožić and M. Kalić, “Aircraft Type Selection Problem: Application of Different MCDM Methods,” in *Advances in Intelligent Systems and Computing*, vol. 572, Springer Verlag, 2018, pp. 156–175. doi: 10.1007/978-3-319-57105-8_7.

[211] B. C. Balusa and A. K. Gorai, “A Comparative Study of Various Multi-criteria Decision-Making Models in Underground Mining Method Selection,” *Journal of The Institution of Engineers (India): Series D*, vol. 100, no. 1, pp. 105–121, Apr. 2019, doi: 10.1007/s40033-018-0169-0.

[212] A. Ishizaka and S. Siraj, “Are multi-criteria decision-making tools useful? An experimental comparative study of three methods,” *Eur J Oper Res*, vol. 264, no. 2, pp. 462–471, Jan. 2018, doi: 10.1016/j.ejor.2017.05.041.

[213] M. Alkahtani, A. Al-Ahmari, H. Kaid, and M. Sonboa, “Comparison and evaluation of multi-criteria supplier selection approaches: A case study,” *Advances in Mechanical Engineering*, vol. 11, no. 2, p. 168781401882292, Feb. 2019, doi: 10.1177/1687814018822926.

[214] S. Vakilipour, A. Sadeghi-Niaraki, M. Ghodousi, and S.-M. Choi, “Comparison between Multi-Criteria Decision-Making Methods and Evaluating the Quality of Life at Different Spatial Levels,” *Sustainability*, vol. 13, no. 7, p. 4067, Apr. 2021, doi: 10.3390/su13074067.

[215] R. E. Hodgett, “Comparison of multi-criteria decision-making methods for equipment selection,” *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5–8, pp. 1145–1157, Jul. 2016, doi: 10.1007/s00170-015-7993-2.

[216] F. Sari, “Forest fire susceptibility mapping via multi-criteria decision analysis techniques for Mugla, Turkey: A comparative analysis of VIKOR and TOPSIS,” *For Ecol Manage*, vol. 480, p. 118644, Jan. 2021, doi: 10.1016/j.foreco.2020.118644.

[217] S. Biswas, “Measuring performance of healthcare supply chains in India: A comparative analysis of multi-criteria decision making methods,” *Decision Making: Applications in Management and Engineering*, vol. 3, no. 2, pp. 162–189, Oct. 2020, doi: 10.31181/dmame2003162b.

[218] R. K. Dewi, B. T. Hanggara, and A. Pinandito, “A Comparison Between AHP and Hybrid AHP for Mobile Based Culinary Recommendation System,” *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 12, no. 1, p. 133, Jan. 2018, doi: 10.3991/ijim.v12i1.7561.

[219] A. Martin, T. M. Lakshmi, and V. Prasanna Venkatesan, “A Study on Evaluation Metrics for Multi Criteria Decision Making (MCDM) Methods-TOPSIS, COPRAS & GRA,” *IIR) International Journal of Computing Algorithm*, vol. 7, no. 1, pp. 29–37, 2018, Accessed: Oct. 31, 2023. [Online]. Available: http://www.ijcoa.com/documents/%20IIR_IJCOA_2018_43.pdf

[220] Z. Wu and G. Abdul-Nour, “Comparison of Multi-Criteria Group Decision-Making Methods for Urban Sewer Network Plan Selection,” *CivilEng*, vol. 1, no. 1, pp. 26–48, Jun. 2020, doi: 10.3390/civileng1010003.

[221] A. Jozaghi *et al.*, “A Comparative Study of the AHP and TOPSIS Techniques for Dam Site Selection Using GIS: A Case Study of Sistan and Baluchestan Province, Iran,” *Geosciences (Basel)*, vol. 8, no. 12, p. 494, Dec. 2018, doi: 10.3390/geosciences8120494.

[222] J. Li, R. He, and T. Wang, “A data-driven decision-making framework for personnel selection based on LGBWM and IFNs,” *Appl Soft Comput*, vol. 126, p. 109227, Sep. 2022, doi: 10.1016/j.asoc.2022.109227.

[223] J. Li, Y. Cheng, T. Wang, and H. Wang, “PLEAS: A Decision Support System for Personnel Evaluation and Selection,” in *2021 4th International Conference on Data Science and Information Technology*, New York, NY, USA: ACM, Jul. 2021, pp. 183–190. doi: 10.1145/3478905.3478943.

[224] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 2007, doi: 10.2753/MIS0742-1222240302.

[225] Hevner, March, Park, and Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, p. 75, 2004, doi: 10.2307/25148625.