



HAL
open science

A smart assessment of business processes for enterprises decision support

Nelson Marcelo Romero Aquino

► **To cite this version:**

Nelson Marcelo Romero Aquino. A smart assessment of business processes for enterprises decision support. Automatic. Université de Lorraine, 2021. English. NNT : 2021LORR0184 . tel-03511735

HAL Id: tel-03511735

<https://hal.univ-lorraine.fr/tel-03511735>

Submitted on 5 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

A Smart Assessment of Business Processes for Enterprises Decision Support

THÈSE

pour l'obtention du titre de

Doctorat de l'Université de Lorraine

(mention Automatique, Traitement du signal et des Images, et Génie Informatique)

par

Nelson Marcelo ROMERO AQUINO

Composition du jury

Président :

- Prof. Henderik Proper, Professeur, Université du Luxembourg, Luxembourg

Rapporteurs :

- Prof. Grégory Zacharewicz, Professeur, IMT – Mines Alès, France
- Prof. Walid Gaaloul, Professeur, Télécom Sud Paris, France

Examineurs :

- Dr. Mario Lezoche, Maître de Conférences, HDR, Université de Lorraine, France
- Dr. Béatrix Barafort, Auditrice Interne, LIST, Luxembourg
- Prof. Hervé Panetto, Professeur, Université de Lorraine, France (Directeur de thèse)
- Dr. Wided Guédria, Chercheure, LIST, Luxembourg; Chercheure Associée, Université de Lorraine, France (Co-directrice de thèse)

This PhD thesis is the outcome of the collaboration between the Luxembourg Institute of Science and Technology (LIST) and the Research Centre for Automatic Control (CRAN) UMR 7039, Joint Research Unit of the University of Lorraine and the French National Centre for Scientific Research (CNRS). This PhD grant has been funded by the LIST.

Acknowledgements

I would like to thank my supervisors Dr. Wided Guédria and Prof. Hervé Panetto for the guidance, support, and words of wisdom they gave me during the Ph.D. I would also like to thank Prof. Grégory Zacharewicz and Prof. Walid Gaaloul for accepting to be the reviewers of this thesis, as well as the other members of the jury, Dr. Mario Lezoche and Prof. Henderik Proper, for examining this work.

Special thanks to Dr. Béatrix Barafort for all the support during these three years of hard work at LIST, it has been a major pleasure to work with such a remarkable professional. I would also like to thank all my colleagues at CRAN and LIST. Specially to the members of the SPG Group: Stephane Cortina, Michel Picard, Samuel Renault, Alain Renault, Phillipe Valoggia. I would also like to thank my friends: Gabriel, Bereket, Nelson, Chantal, Louis, Sarah, Jean-Sébastien, Gabriela and many others for their support and all the good moments we were able to spend together (physically or virtually) during my PhD journey.

I would like to thank my family for their continuous support. I express them my gratitude not only for their support during the Ph.D. but throughout my entire life, staying by my side at each step of my journey. To my father Nelson, my mother Eva, my brother Guido, and my sister Nathalia. I also extend my appreciation and gratitude to the rest of my family members.

Finally, to my beloved wife Kathia, for her unconditional love and support. Thank you for being my companion during both the sunny and the rainy days, for helping me getting through several obstacles and showing me my inner strength during the times in which even I was not able to see it. This accomplishment is also yours. I love you with all my heart.

Nelson Marcelo ROMERO AQUINO | September 2021, France.

“Do or do not.
There is no try.”

Contents

General Introduction	xi
1 Research Context	xi
2 Research Motivation	xii
3 Objectives	xiv
4 Research Methodology	xv
5 Thesis Organisation	xvi

Part I Towards a Smart Assessment Framework

Chapter 1 State of the Art

1.1 Introduction	1
1.2 Enterprise Assessment	1
1.2.1 Types of assessment	2
1.2.2 Maturity and Capability Assessment	4
1.2.3 ISO/IEC 33020 Measurement Framework	5
1.3 Related Work	8
1.3.1 Automated Results Calculation and Presentation	8
1.3.2 Automated Data Collection	9
1.3.3 Frameworks to Enhance Assessments	10
1.4 Research Gaps and Contributions	10

Chapter 2 Characterising Smart Systems

2.1 Introduction	13
2.2 Background	15
2.2.1 General Systems Theory	15
2.2.2 The Concept of Smartness in Different Domains	16

2.2.3	Previous Literature Reviews	17
2.3	Systematic Literature Review and Analysis	18
2.3.1	Systematic Literature Review	18
2.3.2	Papers Analysis	20
2.4	Literature Review Results and Discussion	22
2.4.1	Application domains	22
2.4.2	Publication period	25
2.4.3	Text mining results	26
2.4.4	Smartness and Intelligence	30
2.4.5	Similarities within definitions	31
2.5	Characteristics of Smart Systems	32
2.6	Threats to the Validity of the Literature Review	36
2.7	Smart Capabilities in the Context of Design Science	36
2.8	Conclusion	37

Chapter 3

The Smart Assessment Framework

3.1	Introduction	39
3.2	The Smart Assessment Meta-model	40
3.3	Meta-model Specialisation: Interoperability Assessment	42
3.4	The Smart Assessment Framework (SAF)	44
3.4.1	Data Perception Service	45
3.4.2	Organisation Service	47
3.4.3	Presentation Service	49
3.4.4	Data Flow in SAF	50
3.5	SAF Case Study: Business Process Capability Assessment	51
3.6	The Conceptual Models in the Context of Design Science	53
3.7	Conclusion	53

Part II Smart Assessment Framework Implementations 55

Chapter 4

A Hybrid Approach to Perform Assessments Using Text Evidence

4.1	Introduction	57
4.2	Hybrid Approach to Evaluate Business Processes	59
4.2.1	Overview of the Approach	59

4.2.2	LSTM Model	61
4.2.3	Knowledge Base	62
4.3	Case Study	70
4.3.1	Case Study Description	70
4.3.2	Specialisation for Samples Management Process	71
4.3.3	LSTM Training and Testing	72
4.3.4	Assessment Results	73
4.4	Software Tool	75
4.5	The Hybrid Approach in the Context of Design Science	77
4.6	Conclusion	79

<p>Chapter 5</p> <p>Evaluating the Suitability of Enterprise Modelling Languages to Support Assessments</p>

5.1	Introduction	81
5.2	Enterprise Models and Assessments	83
5.3	Modelling Language and Assessment Requirements Analysis	84
5.3.1	Requirements Decomposition	86
5.3.2	Requirement and modelling elements matching	89
5.4	Analysing BPMN and ArchiMate Considering ISO/IEC 33020	90
5.4.1	Considerations	90
5.4.2	Process Attributes and Indicators of ISO/IEC 33020	91
5.4.3	Results and Discussion	92
5.5	Assessing Capability Level of a Business Process	95
5.5.1	BPMN	97
5.5.2	ArchiMate	102
5.5.3	Discussion	105
5.6	The Enterprise Models Analysis Approach in the Design Science Context . .	106
5.7	Conclusion	106

<p>Chapter 6</p> <p>An Approach to Evaluate Business Process Capability Using Process Models</p>
--

6.1	Introduction	109
6.2	Previous Attempts to Evaluate Process Models	110
6.3	GCN Approach for Business Process Models Assessment	111
6.3.1	Process Model Generator	112
6.3.2	Process Model Classifier	114

6.3.3	GCN Deployment	115
6.4	GCN Approach Experiments and Results	116
6.4.1	System Setup	116
6.4.2	Dataset	116
6.4.3	Results	116
6.5	The GCN Approach in the Design Science Context	118
6.6	Conclusion	118
General Conclusion		121
Bibliography		125
Appendices		149
Appendix A		
Enterprise Modelling		
A.1	Modelling Languages	151
A.2	Enterprise Modelling Languages	153
A.2.1	Business Process Model and Notation (BPMN)	155
A.2.2	ArchiMate	156
Appendix B		
Deep Learning		
B.1	Long Short-term Memory (LSTM)	160
B.2	Word Embedding	161
B.3	Graph Neural Network (GNN)	162
Appendix C		
Knowledge-based Systems		
C.1	Ontologies	163
C.2	Inference Engines	163

List of Figures

1	Overview of the three-cycle view of the Design Science Research method followed in this work (adapted from [49]).	xvi
2	Organisation of this thesis, linking each chapter and its content with the tasks. . .	xvii
1.1	A brief overview of the elements of an assessment, specialised for business process capability assessment.	5
1.2	A brief overview of the elements of an assessment specialised for business process capability assessment.	6
2.1	Overview of the methodology followed in this work.	18
2.2	Overview of the methodology to explore the definitions through text mining techniques.	21
2.3	Domains of the papers that presented definitions of Smartness or Smart Systems.	24
2.4	Number of papers containing definitions per year.	26
2.5	Number of the papers that presented definitions of Smartness or some Smart System per year considering the domain.	27
2.6	Number of occurrences of the top-25 most repeated words considering the full text of the definitions.	28
2.7	Number of occurrences of the top-25 most repeated keywords that were manually extracted from the definitions.	29
2.8	Visualisation of the embedding vectors using the 25 most words in the definitions.	30
2.9	Weights of the most frequent words in the definitions considering their appearance in definitions of each domain.	33
2.10	Meta-model of Smart Systems.	35
3.1	Meta-model defining the components of a Smart Assessment.	41
3.2	Smart Assessment meta-model specialised for Enterprise Interoperability.	43
3.3	Conceptual framework defining the components of a smart assessment.	44
3.4	Flow of the assessment data through each component of the Application layer of the SAF.	50
3.5	Implementation example of the SAF for the assessment of a Smart Samples Management (SM) process using the TIPA framework as Assessment Framework.	52
4.1	Overview of the approach followed in this work.	60
4.2	Pipeline of the LSTM approach and the architecture of the network.	62
4.3	The Assessment Ontology.	63
4.4	The Assessment Ontology specialised for Process Capability Assessment.	64
4.5	The Process Assessment Ontology based on the ISO/IEC 33020 standard in detail.	65

4.6	Base classes of the ontology used in this work represented through OWL.	66
4.7	Process-specialised classes of the ontology used in this work represented through OWL.	67
4.8	References used in each level of the PCAO.	68
4.9	Base Practices added for the Samples Management Process, including the already existing Generic Practices.	71
4.10	Train and test loss during training of the model.	73
4.11	Organiser component pipeline and the results for the SM process.	74
4.12	Architecture of the system. Elements represented with white boxes are concrete components providing Application Services and accessing Data Objects, whilst the light dotted lines and boxes are used to add information regarding a specific component. Arrows with white triangles represent inheritance.	75
4.13	Overview of the tasks performed by the Front-End and the Back-End applications of the software tool.	76
4.14	Screens of the tool that are used during an assessment.	78
5.1	Relationship between the Requirement of the assessment framework and the Element of the modelling language.	85
5.2	Overview of the evaluation process.	86
5.3	Overview of the requirements decomposition and formalisation through the Pseudo-Requirement Graph-based approach. Dotted lines and boxes are used to describe the properties of the output artefacts of the activities of the model.	87
5.4	Example of the decomposition of a requirement <i>Rq</i>	88
5.5	The Goal Question Metric paradigm within the scope of the proposed approach.	89
5.6	Results obtained through the analysis of the BPMN and ArchiMate modelling languages following the proposed methodology.	97
5.7	BPMN model describing the Samples Management Process.	98
5.8	ArchiMate model describing the Samples Management Process.	102
5.9	Ratings given to the Indicators of the Process Attributes 2.1 and 2.2 of the ISO/IEC 33020 for the Samples Management Process, considering the ArchiMate model.	104
6.1	Overview of the methodology followed in this work.	112
6.2	Example of a visual representation of an input process model and the network architecture.	115
6.3	Example of a correct process model of the dataset.	117
6.4	Evolution of the loss of the network for the train and test sets throughout the epochs.	117
A.1	Elements of a modelling language and other related aspects (adapted from [308]).	152
A.2	Framework of ArchiMate (adapted from [237]).	156
B.1	Typical pipeline of a Deep Learning model.	159
B.2	An LSTM memory cell.	160

General Introduction

1 Research Context

The world is evolving towards a new era based on the use of renewable energy and smart technologies [1]. This third industrial revolution is a journey towards the extended exploitation of clean sources of energy and the capabilities of smart technologies [1]. From a technological perspective, this revolution is mainly grounded on a digital transformation [2] in which there is a strong focus on the integration of technology into all areas of the public and private organisations to increase productivity and value creation [3]. This paradigm is centred on the use of ubiquitous elements such as Cyber-Physical Systems [4, 5], Internet of Things (IoT) [6, 7] devices, etc., that allow to generate, consume, and process data, in order to act on a certain environment or provide the necessary information for decision-making [8].

Enterprises are in the centre of this novel ground-breaking industrial era, as they need to adapt to the new challenges derived from this new scenario in order to deal with aspects such as sustainability, regulatory compliance, effective adoption of novel technologies, cost optimisation, effective risk management, etc. These adaptations need effective and, in some cases, considerable transformations that require an analysis of their capabilities and limitations as a first step before planning any transformation initiatives. For this purpose, conducting an assessment of the enterprise is essential. If done correctly, it allows to analyse strengths and weaknesses of enterprise capabilities [9], allowing decision-makers to understand the As-Is state, and consequently plan a targeted To-Be state. Assessments can be qualitative or quantitative. Qualitative assessments are focused on non-measurable aspects and it is focused on more subjective measures [10], whilst quantitative assessments put emphasis on measurable attributes from a more objective perspective, usually they are centred on measuring performance [11].

The novel technology-centric paradigm introduced in the context of the industrial revolution modifies not only enterprise aspects concerning operations with clients, providers, and partners, but also those related to internal facets such as the production pipeline, communication process, collaboration between teams, definition of new strategies, etc. Moreover, this scenario fosters the generation of large amounts of heterogeneous data including text, images, videos, audio, raw sensor data, enterprise models, etc., providing information of the enterprises from different perspectives. This available data can be used as evidence to perform different types of assessments. However, the large variety of types of data demand novel assessment methods and systems that could be capable of processing them to obtain assessment results.

In this context, besides providing novel, fast and efficient ways to generate data allowing to describe internal and external elements of an enterprise, disruptive technologies also provide innovating alternatives to perform assessments, using the available data to obtain reliable views of the As-Is state of enterprise entities that can be used for decision-making purposes. In this sense, the Artificial Intelligence (AI) [12] field includes a particularly relevant set of techniques that gained popularity during the recent years due to their capability to achieve state-of-art

results for different tasks. In an industrial context, AI methods allow to effectively support decision-making and, if required, act in increasingly complex scenarios. AI approaches can be viewed from two perspectives: data-driven and model-driven. In data-driven AI, the knowledge of the system is automatically inferred from historical data. It includes a sub-field named Deep Learning (DL) [13], which is widely extended nowadays in both the industry and academia. DL is a form of Machine Learning (ML) that enables computers to learn from past experiences and build an understanding of certain aspects of the world through a hierarchical representation of concepts [14]. DL techniques are characterised by the need of data to train models to perform specific tasks such as speech recognition, image segmentation, video classification, etc. On the other hand, a more classical AI perspective comprises techniques that are symbolic and strongly oriented to the use of rules and pre-defined heuristics to perform certain tasks. Fuzzy logic [15] and expert systems [12] are two examples of classical AI approaches. In this thesis, these methods are considered as part of the model-driven AI category, since the knowledge must be manually modelled by human experts.

These novel data processing methods and technologies could provide means to automatically manipulate and process the assessment evidence, which is any type of data reflecting the state of an assessed entity [16]. The assessment process is typically performed through the following steps: planning, data collection, results determination, and results presentation [17]. In particular, performing the data collection and results determination phases of an assessment often demands an investment in terms of time and effort if they are carried out by a team of human assessors. The data collection step refers to all tasks related to gathering evidence that allows to describe the assessed entity, whilst the results determination step comprises the activities of analysis of the evidence to provide a view of the As-Is. Purely manual assessment initiatives are often resource-demanding and they are more likely to produce incomplete, more subjective, and even biased assessment results, considering that some evidences may be incorrectly collected, processed or analysed by the responsible of performing those tasks. Therefore, automating the assessment process (or some of its activities) through the use of dedicated systems could provide high value for enterprises. The use of a system able to perform a fully automated assessment has three main advantages: the first point is that the need of collecting, pre-processing and filtering the relevant data must not be necessarily done by humans, resulting in a significant reduction of resource consumption to perform the assessment. Moreover, if the human assessor only needs a pre-processed evidence, the assessment system could stop the process in that step, giving the expert the liberty to make its own decisions using that previously treated evidence. The second point is that the system could provide final assessment results that could be used directly by the stakeholders, or as an alternative second opinion to be considered by a human decision maker. The latter option could be specifically useful when performing qualitative assessments. Finally, the third point is that automated appraisals reduce the probability of obtaining biased, incorrect, incomplete or too much subjective assessment results.

2 Research Motivation

Throughout the years, several works have aimed at improving the assessment process through the exploitation of technologies and methodologies that meant novelty or disruption at a certain point in time [18–35]. Researchers and industry practitioners have proposed and tested different approaches to tackle the problem of reducing human effort when performing assessments (qualitative and quantitative), while ensuring reliability and efficiency. Those approaches can be grouped in three main categories: manual, semi-automated, and context-aware assessments.

Manual assessments completely rely on the participation of human experts to plan the appraisal, gather and analyse the evidence, and define and communicate the results. Research and industry efforts regarding manual assessments are often focused on proposing frameworks [18–26] to facilitate the study of the assessment evidence or providing formal mechanisms to plan and perform assessments. Regardless of the specific methodology or framework used to perform an assessment, the process consists of a series of steps that are common to all appraisals: assessment planning, data collection, results determination, and results communication [17]. In a full manual assessment, all of these steps are performed by human experts. Semi-automated assessments put emphasis on the use of computers to automate certain tasks of the process [36]. However, the participation of humans to perform some activities is still required. In general, works addressing semi-automated assessment introduce tools to aid during the appraisal through the automation of its results calculation phase and providing user interfaces to upload assessment evidence or partial results that need to be further aggregated to obtain a final result. The works proposed by [27–30] are examples of such software tools. The automatic generation of assessment reports is a common feature of these tools. Finally, context-aware assessments rely on context-aware systems [37] that are focused on the use of physical or virtual sensors [37] to gather data about the assessed entity. The collected data is then passed to context manager able to process and reason using the data to obtain a modelled representation of the context. This context, which is defined in [38] as "any information that can be used to characterise the situation of an entity", is later distributed to the consumers that are interested [37]. There are different ways to deal with context-aware assessments. For qualitative assessments, for instance, Process Mining techniques [39, 40] can be employed to use listeners that could be connected to Enterprise Resource Planning (ERP) systems to collect system logs in order to derive business process models that can be afterwards analysed to assess the process. Some processing could also be performed (on the event logs or on the obtained models, using domain knowledge for instance) to obtain business process models structured with a more specific purpose or with a certain characteristic [31–33]. For quantitative assessments, context-aware assessments could be based on the use of sensors perceiving an industrial environment to obtain data in real-time that can be afterwards pre-processed and shown through visualisation tools [34, 35] to stakeholders in order to make decisions.

The approaches that were mentioned before are part of the different research initiatives addressing the improvement of the assessment process by proposing a certain strategy, methodology, artefact, or technology. These approaches are generally focused on concrete scenarios and domains. Moreover, when considering the level of automation of the assessment process, there is a gap between semi-automated and context-aware assessment. The former seems strongly focused on automating the results calculation phase of the assessment, whilst the latter is centred on automating the data acquisition and data modelling activities. This gap represents the absence of fully automated assessment approaches able to input data and provide result with autonomy in order to reduce human effort and provide reliable assessment results. In this sense, there is a need for a holistic framework able to consider the automation of assessments from an end-to-end perspective. This framework could be used by researchers and industry practitioners as a guide for the development of assessment systems capable of performing fully automated assessments, thus closing the gap between the automation of the data collection and results determination phases of the appraisals. Additionally, the framework must be generic enough to be able to guide the design and development of assessing systems with independence of the algorithms or technological resources used by them. In this sense, genericity has a set of advantages, for instance, it allows the framework to be instantiated for multiple domains by providing clear and simple descriptions of its elements and the basic functionality of the system to be developed without

the need of specific domain-knowledge [41], it fosters re-usability of the components of the developed system in different applications [42], it is independent from the technological resources thus giving the system designers the liberty to select the technology that they consider is the best fit for their problem, and it allows to analyse if an assessment system that was previously developed for a certain application domain has elements that act as instances of some elements of the framework using some similarity measure [43], which avoids to re-invent technologies, approaches or methods.

Note that an assessment system developed using the framework must not only be capable of automating certain calculation tasks or consuming and modelling context data, but it has to encompass a set of capabilities to ensure effective end-to-end assessments. In this sense, the capabilities of smart systems such as knowledge, reasoning, learning, communication, or perception [44] could serve for such purpose, since they allow them to continuously perceive a certain environment, make inferences regarding the state of that environment, have communication between the system elements, and perform specific actions if it is required [44]. Therefore, the framework could include these capabilities to guide the development of smart assessment systems that could be capable of performing automated end-to-end assessments. Moreover, the concept of smartness in the context of enterprise assessment could not only imply that the assessment result can be defined properly by the system without human involvement, but also that some actions could be taken once the result has been calculated so as to initiate the transformation process towards the achievement of the To-Be state instantaneously. Indeed, mechanisms to automatically perform actions to modify the environment and react to new states of that environment could also be used once the assessment result has been obtained [45], thus closing a cycle composed of sensing the environment, thinking and deciding about possible actions to take given its state, and actually performing those actions. Although this is not required during an assessment since appraisals are only focused on having a diagnostic of the assessed entity, it is a capability that is inherent of smart systems and it could also be taken into account by system designers if they consider necessary. On the other hand, not only external elements from the environment could be modified, but also internal components of the assessment system. Specifically, the learning capability of a smart system could provide the possibility to adapt to new challenges or improve its own performance, so as to provide more reliable assessment results as the system evolves using or visualising novel data.

Considering the research context and motivation, this thesis aims at addressing the lack of a generic framework capable of guiding the design and development of automated end-to-end assessment systems in enterprises.

3 Objectives

The objective of this thesis is to **propose a framework to perform smart assessments of enterprise entities: the Smart Assessment Framework (SAF)**. The hypothesis of this work is that a conceptual framework grounded on the concepts of smart systems can provide efficient support for the development of systems that allow to perform automated assessments in enterprises.

Defining such framework implies a set of challenges that must be addressed: (1) An analysis of the smartness concept as well as the characteristics of smart systems. This will be supported by an extended analysis of the literature; (2) The framework for smart assessment must be designed considering both the characteristics of smart systems and the activities of the assessment process; (3) The validation of the framework must be done considering different assessment scenarios.

The scenarios could be based on real-world use cases and defined in different ways to cover various aspects such as the assessment evidence that must be treated (structured, unstructured, etc.), the characteristic to be analysed (maturity, capability, interoperability, risk, etc.), and the assessment type (qualitative or quantitative). The use case addressed in this thesis is focused on business process capability assessment scenarios. Business processes are vital due to that they comprise the set of activities or business functions that are performed following a certain sequence in order to obtain results that provide value to the organisation [46]. The relevance of understanding the state of business processes and constantly improve them lies on the fact that most of the problems related to efficiency and effectiveness in an organisation are originated in non-mastered processes [22].

Given the objective of this work and the research challenges enumerated before, the focus is set on answering the main Research Question (RQ):

What are the elements that compose a smart enterprise assessment framework?

To address this RQ, the main objective of this thesis addresses the need of a framework grounded on the capabilities of smart systems to guide the development of tools, approaches and methodologies to perform smarter enterprise appraisals. In this sense, the framework must be first properly defined and afterwards validated. The secondary objectives of this work, which are introduced next, are devised with the goal of addressing those two requirements:

- Design a framework to perform smart assessments of enterprise entities.
- Validate the framework considering different assessment scenarios.

The first specific objective is related to the definition of the framework. For the sake of clarity, a conceptual modelling approach is used for the definition of the framework, this allows to enhance the understanding of its architecture by making visible its relevant features, characteristics and complex relationships between its elements [47]. Regarding the validation of the SAF, this thesis is focused on qualitative assessment of business processes. Specifically, scenarios addressing capability evaluation of a chemical samples management business process are targeted. The validation is therefore based on the implementation of systems in charge of supporting appraisals for those scenarios. This is achieved by instantiating the elements defined by the framework. The main difference between the scenarios is the type of assessment evidence used for the assessment. In this sense, two types of evidences are considered for the implementations: textual process descriptions and business process models. Considering the latter type of evidence, a relevant aspect when performing capability assessment using enterprise models is that the information that can be found in such type of artefacts depends on the modelling language that is used to design them. In this sense, understanding which assessment requirements can be answered by the models that are designed with a certain language is a vital step during the development of a system devised for the automatic treatment of those models. This is an issue that is also addressed in this thesis.

4 Research Methodology

The SAF is considered as an artefact that must be developed with the purpose of improving a specific environment. Considering this perspective, the Design Science Research (DSR) method [48] is followed in this work. DSR aims at improving an environment by incrementally developing artefacts (and the processes to build those artefacts) following a specific design cycle. The cycle

is based on the feedback of the environment, scientific theories, human experience and expertise, and meta-artefacts within a knowledge base. Specifically, the three-cycle view of DSR [49] is considered for the development of the contributions of this thesis, which are viewed as research artefacts in the context of DSR. Figure 1 presents an overview of the three-cycle view of DSR.

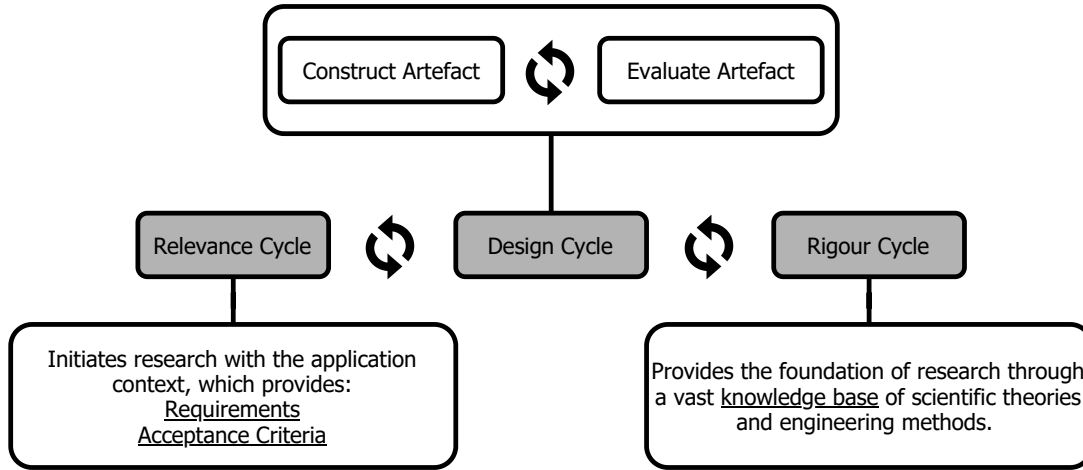


Figure 1: Overview of the three-cycle view of the Design Science Research method followed in this work (adapted from [49]).

The **Design Cycle** is composed of two core activities: the construction of the research artefacts and processes, and their evaluation. This cycle is based on performing iterations that include feedback between those activities in order to reach a refined and satisfactory research artefact, with proven capability to improve a targeted environment. This cycle is connected, through an iterative linkage to the two other research cycles of the methodology: the Relevance Cycle and the Rigour Cycle.

The **Relevance Cycle** provides the application context of the research. It serves as a bridge between the contextual environment of the research project and the design science activities [49]. This cycle provides the Requirements that the research artefact must fulfil in order to make the improvement on the targeted environment, as well as the Acceptance Criteria to be used as evaluation metrics during the evaluation of the artefact. Note that the evaluation of the artefact will determine if further iterations are needed, if requirements that are considered in the project are incomplete or incorrect, or if the evaluation metrics are not suitable for the artefact. Therefore, there is a loop between this cycle and the Design Cycle in Figure 1.

The objective of the **Rigour Cycle** is to establish a link between the design science activities and the knowledge base used to do research. This base is composed of scientific foundations, human experience and expertise that define the state-of-the-art in the application domain of the research, and the existing artefacts and meta-artefacts (also known as the processes for artefact development) that exist in that application domain.

5 Thesis Organisation

The contributions of this thesis are grouped in two parts, as shown in Figure 2. Part I aims at accomplishing the first specific objective related to the design of a framework to perform smart assessments. Therefore, it is focused on the definition and presentation of the SAF. Part II describes concrete implementations of the SAF to perform automated appraisals, thus aiming at

accomplishing the second specific objective related to the validation of the framework considering different assessment scenarios.

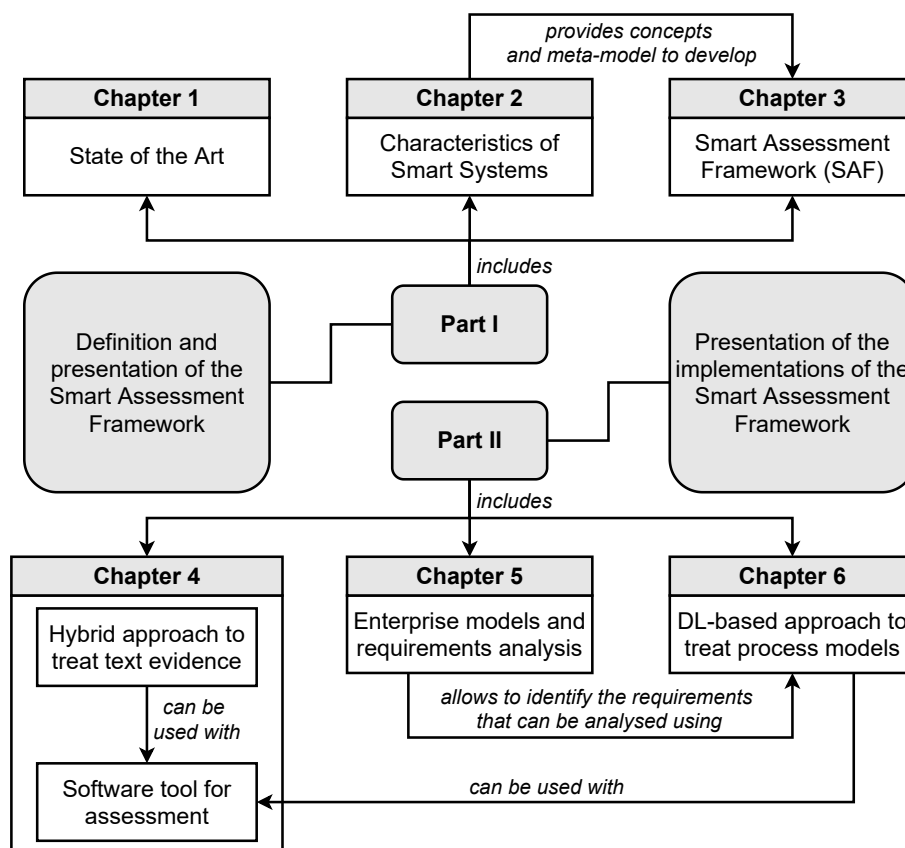


Figure 2: Organisation of this thesis, linking each chapter and its content with the tasks.

Considering Part I, **Chapter 1** introduces the theoretical aspects that are addressed throughout this thesis. It also presents the main related work pointing out the contributions of this thesis considering the research gaps in the literature. The research methodology that guides the development of this work is also described in this chapter. On the other hand, since the SAF is intended to be strongly grounded on the characteristics of smart systems, **Chapter 2** presents a systematic literature review that aims at analysing definitions of such type of systems independently from their application field (e.g. Smart Cities, Smart Homes, or Smart Industries). As a second step of the analysis, the concepts that could allow to obtain a generic characterisation of smart systems are automatically extracted using natural language processing techniques [50]. Finally, **Chapter 3** describes the SAF from a conceptual point of view by proposing conceptual models for smart assessment while considering specific assessment scenarios.

Regarding Part II, **Chapter 4** introduces an approach focused on the use of text data as assessment evidence to perform business process capability evaluation. It is based on the use of a combination of data-driven (Deep Learning) and model-driven (Ontology) AI methods to automatically calculate ratings for a business process and define process capability level. The chapter also presents the description of the architecture of a software tool that integrates the approach mentioned before so as to be used by a final user through a web application. On the other hand, **Chapter 5** drifts from the use of text data as assessment evidence and considers the use of enterprise models instead to be used as input for a smart evaluation system. The

chapter presents a manual approach to define which requirements can be answered or evaluated when performing assessments using enterprise models as evidences. Finally, **Chapter 6** describes an approach based on a type of neural network named Graph Neural Network (GNN) to automatically analyse business process models in order to provide an initial rating to be used for capability level calculation from a data-driven AI perspective.

Table 1 presents the content of each cycle within the context of the DSR method introduced in the previous section, organised by the artefacts that are proposed in this thesis. It also shows the chapters in which the artefacts are introduced. The main requirements for the artefacts are shown in the Relevance Cycle column, whilst the sources of knowledge are presented in the Rigour Cycle column. On the other hand, the Design Cycle is decomposed to show the descriptions of the Construction and the Evaluation activities. It is worth mentioning that the objective of this table is only to summarise the content of the main chapters of this thesis from the perspective of the three-cycle view of DSR method. It is, therefore, not intended to provide an extended nor detailed description of each proposed artefact and the process that is followed for their development, these aspects are fully addressed in their corresponding chapters.

Table 1: Content of each cycle within the DSR methodology considering the artefacts proposed in each chapter of this thesis. The Construction and the Evaluation columns correspond to activities of the Design Cycle.

Chapter	Artefacts	Relevance Cycle	Rigour Cycle	Construction	Evaluation
I-2	Conceptualisation of the smart capabilities. Meta-model of smart systems.	Set of characteristics of smart systems.	Papers containing definitions of smart systems.	Systematic Literature Review. Natural Language Processing Techniques. Papers Analysis.	Threats to validity analysis. Concepts derived quantitatively.
I-3	The Smart Assessment meta-model. The Smart Assessment Framework.	A framework to guide the development of smart assessment approaches.	Characteristics of smart systems, international standards, literature regarding assessment, literature addressing novel technologies.	Conceptual modelling perspective.	Illustrative case studies. Concrete implementations of the SAF in the rest of the chapters.
II-4	Hybrid approach to automatically treat text evidence.	Combination of model and data-driven approach for the automatic treatment of text evidence.	Literature addressing ontologies, knowledge-bases, existing ontologies, neural networks, and natural language processing.	Knowledge-base building. Definition of rules. Definition and training of the architecture of a neural network.	Case study using real-world text evidence to perform business process capability assessment.
II-4	Software tool for assessment.	A web application to allow users to perform automated assessments.	Documentation of the framework for web development, literature addressing web application architectures, documentation of the web server, documentation of the database, documentation of the DL framework.	Software development using the components of SAF to define the architecture of the system.	Case study using real-world text evidence to perform business process capability assessment, implementing the hybrid approach from the same chapter. Feedback from assessment experts.
II-5	Enterprise models and assessment requirements analysis framework.	A methodology to define elements of enterprise models that are able to answer assessment requirements defined in an assessment framework.	International standards for assessment, enterprise modelling languages papers and documentations, literature addressing the Goal-Question-Metric methodology, literature addressing Requirements Engineering.	Development of a framework combining different analysis methodologies related to requirements engineering and enterprise modelling language assessment.	Case study to analyse two different modelling languages and one international standard. Case study based on performing capability assessment using the proposed framework.
II-6	DL-based approach to treat process models.	Automated treatment of business process models through a data-driven approach.	Literature addressing graph neural networks, literature addressing business process models, documentation of the DL framework.	Definition and training of the architecture of a neural network to treat process models as input data.	Case study using real-world and synthetic data to perform business process qualitative assessment using process models as input evidence.

Part I

Towards a Smart Assessment Framework

State of the Art

Contents

1.1 Introduction	1
1.2 Enterprise Assessment	1
1.2.1 Types of assessment	2
1.2.2 Maturity and Capability Assessment	4
1.2.3 ISO/IEC 33020 Measurement Framework	5
1.3 Related Work	8
1.3.1 Automated Results Calculation and Presentation	8
1.3.2 Automated Data Collection	9
1.3.3 Frameworks to Enhance Assessments	10
1.4 Research Gaps and Contributions	10

1.1 Introduction

This chapter presents the theoretical aspects that are relevant to understand the content of this thesis. The concepts that are introduced, specifically in Section 1.2, are related to enterprise evaluation including the types of assessments, an explanation of maturity and capability assessment, as well as a brief description of the ISO/IEC 33020 Measurement Framework, which is an international standard related to business process capability assessment. On the other hand, this chapter also introduces the related work in Section 1.3 with emphasis on the attempts to automate the assessment process and the frameworks introduced in the literature to improve or guide appraisals. Finally, Section 4 details the main components of the Design Science Research methodology, which is the core research method used in this work.

1.2 Enterprise Assessment

An assessment is the act of judging the value, quality or importance of one or more aspects of an entity [51]. Specifically, in enterprises, assessments are focused on evaluating structural or behavioural elements from some perspective such as maturity, interoperability [24], agility [52], technological readiness [53], regulatory compliance [54], etc. Enterprise assessments are composed of four main steps according to the ISO/IEC 33002 standard [17]: planning, data collection, results determination, and results presentation.

During the planning phase, the activities that will be performed during the assessment are defined [30, 55, 56], the objectives of the stakeholders are identified [19], as well as the roles and responsibilities [56]. The main objective of this phase is to establish the grounds upon which the evaluation will be performed. Once the plan is defined, the next activity consists of gathering the required data for the assessment. Depending on the needs of the assessment, data could come from different sources such as interviews [19, 55, 57, 58], questionnaires [59, 60], surveys [58, 61], workshops [62], documentation of the organisation [19, 57], process models [63], data from a Business Process Management System (BPMS) [64], etc. Given the variety of possible data sources and data collection methods, part of the collected data may not be valid for the assessment due to different factors. Therefore, it must be validated, which is done by confirming that the data is representative enough of the assessed entity, as well as checking its consistency, objectivity, and sufficiency [17].

With the available data obtained in the data collection and validation phase, the results of the assessment are determined. This depends on the type of assessment, for quantitative assessment, the proper numerical calculations must be performed using the preferred calculation method. This includes the use of statistical and mathematical approaches to calculate Key Performance Indicators [11], and in some cases, more sophisticated methods such as Clustering [58], Artificial Neural Networks [28], Ontologies combined with inference engines [65], etc. For purely qualitative assessments, the assessment team must provide feedback considering the collected evidence and rate the quality indicators for the assessed entity. If each assessor provides its own rating, these must be aggregated [66]. Note that qualitative assessment may also use hybrid approaches to provide a final assessment result [67]. This is usually done by gathering subjective data that is highly dependent on the experience and knowledge of the assessors, and process this data using statistical methods in order to reduce this subjectivity to provide a robust assessment result. Methods such as Fuzzy theory [24], Analytic Hierarchy Process (AHP) [68], and Fuzzy AHP (FAHP) [69] are usually used in this phase of the assessment.

As for the results presentation phase, the results of an assessment can be descriptive, prescriptive or comparative [70]. Descriptive results aim at providing a diagnostic of the state of the assessed entity [71, 72], and they do not provide any improvement recommendation. Quantitative assessments tend provide these types of results since they simply describe the As-Is state; however, qualitative approaches such as maturity models may also provide only descriptive results [70]. Prescriptive results, on the other hand, provide guidelines on improvement measures [71] by detailing specific courses of action [72]. Finally, comparative results are aimed at serving for internal or external bench-marking [73].

1.2.1 Types of assessment

Assessments can be classified considering different aspects. This section presents the types of assessment considering: the use of levelling methods, the measuring mechanisms, the nature of the assessed entity in terms of openness, and the readiness of the entity to have a certain functioning.

Use of levelling methods

Levelling-based assessment, also known as maturity assessment, is based on the use of maturity models that reflect aspects of the reality or capabilities of an entity by defining qualitative attributes that are used to qualify it into one or more clearly defined classes [74]. Considering business processes, maturity assessment offers an effective way to measure them within an organ-

isation [75]. The maturity represents the extent to which processes are explicitly or consistently deployed by an organisation [76]. In this context, maturity models conceptually represent levels of increasing capability of a maturing element, in order to evaluate its advances considering some defined focus areas [74]. Since they use of levels as evaluation result, maturity assessment approaches are considered levelling methods [24]. On the other hand, non-levelling methods, also known as non-maturity model-based methods, are a diverse group of approaches that are not generalised for defining a maturity level but they are rather focused on defining some numerical or categorical value as assessment result.

Measurement Mechanism

Considering the mechanism to measure the assessed entity, there are two groups: qualitative and quantitative. Qualitative assessments are mostly based on the use of subjective measures to evaluate the entity, generally they are based on a rating scale composed of linguistic variables (e.g. “Regular”, “Good”, “Optimised”) [77]. This measuring mechanism is more commonly used in maturity-based methods. On the other hand, quantitative assessment methods define numeric values to provide a final representation of the value of the assessed entity. Quantitative assessment is often focused on measuring performance that could address finance, customer, supplier, process, time, learning and growth, innovation, or employee [11]. However, this it is not always the case. For instance, the interoperability of an enterprise could be also described through quantitative mechanisms, e.g. using a rating scale from 0 to 100% to explain the capability of the enterprise to interoperate with another one [10]. Note also that quantitative mechanisms could comprise different aggregation methods to define assessment results including operations such as the max, median, min, average, weighted average, etc.

Openness of the assessed process

Regarding the openness of the assessed entity, the literature [24] considers two types of assessment, which were inherited directly from the general systems theory [78]: white-box and black-box methods. White-box methods focus the analysis of the internal structure of the evaluated system. In this sense, the input-output mappings are considered, as well as the transformation structure and the state of the system [79]. Black-box methods are focused on analysing only the inputs and outputs of the system, without addressing aspects such as its internal properties or interactions.

Readiness of the assessed entity

Assessments can also be classified considering the point in time in which the appraisal is performed. The *a priori* assessment, also known as *ex-ante* evaluation, is applied to assess the potential of the assessed entity [24]. For instance, when performing interoperability assessment, the evaluation is centred on the capability of an enterprise to interoperate with one or more organisations considering elements that are the object of the interaction such as data, services, processes and businesses [24]. On the other hand, the *mid-term* assessment is performed during the operation of the assessed entity. As an example, a business process could be subject of a mid-term assessment to evaluate performance, resources consumption, quality of the activities, achievement of sub-objectives, etc., at real-time in order to take corrective or quality-enhancement actions if needed. Finally, the *a posteriori* or *ex-post* evaluation is done with the objective to evaluate performance or quality after the entity has finished operating [24]. Considering the interoperability evaluation example, the appraisal would be focused on analysing if the

interaction between the organisations has been effective, efficient, compliant to standards, etc., once that interaction has finished.

1.2.2 Maturity and Capability Assessment

Maturity assessment in enterprises is focused on using maturity models that define certain attributes that are used to categorise an entity into one or more classes [74]. The results of a maturity assessment is mainly used to identify strengths, weaknesses, and possible improvement points for the assessed entity. It also gives a way to prioritise these improvements depending on the targeted maturity level. When performing maturity assessment, maturity models represent frameworks describing conceptual levels of increasing capability of a maturing element in order to evaluate its advances considering some defined focus areas [74]. Research and industry practitioners have defined a large set of maturity models to address several enterprise aspects throughout the years such as: Capability Maturity Model [80], Capability Maturity Model Integration (CMMI) [18], Project Management Maturity Model [81], e-Learning Maturity Model [53], Digital Readiness Maturity Model [82], Maturity Model for Enterprise Interoperability [24], Information System Interoperability Maturity Model [83], among others.

From a business process-related perspective, maturity assessment also represents an effective way to measure the quality of business processes that are deployed by an organisation [75]. In this context, the maturity of the enterprise is measured evaluating all of its processes, grouped within a specific area of focus. On the other hand, the capability represents the extent to which a single process is explicitly or consistently deployed by the organisation. Therefore, if a set of processes is the object of analysis, the maturity of the organisation is being measured [76]. Otherwise, if only one process is assessed, the assessment is focused on the capability of that process. This means that measuring the maturity of the enterprise often implies to perform capability assessment of a set of processes and aggregate their results to reach a definition of the maturity level. The result of a capability assessment is often represented through a capability level, which is a well-defined evolutionary plateau that describes the capability of an enterprise in relationship to a certain process area [18].

In maturity and capability evaluation, assessment frameworks are relevant elements that take part when performing appraisals. An assessment framework provides a schema that is used to portray some qualitative or quantitative characteristic of an entity [16]. It includes a set of generic requirements that the assessed entity must fulfil. It also includes the results calculation and determination methods to be used for the definition of the assessment results. Moreover, it could also define a set of improvement recommendations that could be linked to the requirements in case they are not accomplished by the entity. It is worth mentioning that some of the approaches introduced in the literature use different names such as assessment method or measurement framework. However, all of them include the same elements, thus this thesis treats them as assessment framework in order to have unified naming. On the other hand, there is no clear and unanimous definition of the content of an assessment framework and a maturity model in the literature. Some maturity models already include elements that are typically defined by assessment frameworks, and the opposite also occurs, in which the framework establishes maturity levels and the attributes of the assessed entity without using a maturity model as reference material. Notwithstanding, this thesis considers them as distinct and complementary elements. Some relevant assessment frameworks in the industry and academia are: ISO/IEC 33020 Measurement Framework [20], Standard CMMI Appraisal Method for Process Improvement (SCAMPI) [84], TIPA Framework [85], Smart Manufacturing Maturity Assessment Framework [86], among others.

The concepts that are related to assessments (and the manner they are connected) can be graphically described through meta-models. Figure 1.1 presents some of the main elements of an assessment using a meta-model. The elements shown in the figure are based on the meta-model presented in [87]. The model is not intended to reflect a full view of the assessment process, but it allows to show how a Capability Assessment is linked to the Requirements that are defined by an Assessment Framework. In this sense, a Requirement defines a need of a stakeholder regarding the evaluated attribute [88]. Regardless of the assessment scope, in all cases there is a set of Requirements that define the Quality Attribute to be evaluated during the appraisal.

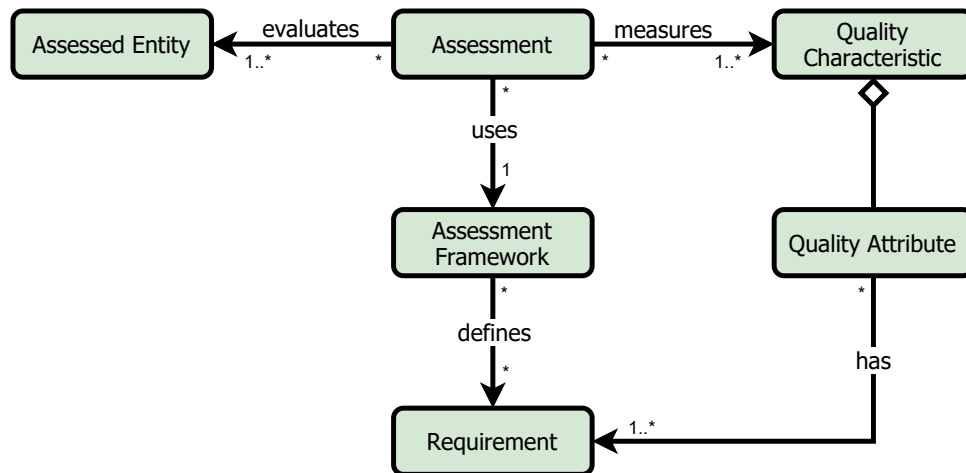


Figure 1.1: A brief overview of the elements of an assessment, specialised for business process capability assessment.

Note that Figure 1.1 shows the link between the Assessment Framework and the Requirement element. In this sense, a framework can define one or more requirements, which contain a set of criteria that the Assessed Entity element must fulfil in order to reach a certain level of quality or compliance to some regulation or standard.

1.2.3 ISO/IEC 33020 Measurement Framework

Although this thesis introduces approaches that can be applicable to evaluate different types of enterprise entities and aspects, all case studies of this work are related to business process assessment. In this context, the ISO/IEC 33020 [20] standard was selected as main Assessment Framework to guide the appraisals.

ISO/IEC 330xx standard series is a well-known and established international standard that is widely used in the literature and the industry. It is composed of a set of standard documents describing the requirements for performing process assessment in organisations, requirements for process measurement frameworks, requirements for process reference, process assessment and maturity models, a guide for process improvement, a process measurement framework for assessment of process capability, assessment models (e.g. for software testing, system life cycle process, safety extension, etc.), and a set of definitions of concepts and terminology that are relevant in the scope of the standard.

Specifically, the Measurement Framework (ISO/IEC 33020) defined by the standard contains a set of Process Capability Levels in which each level groups one or more Process Attributes (PA). A PA represents a measurable property of process capability and it implicitly defines a set of Indicators or Requirements to evaluate a business process. It also provides a result calculation

method. Figure 1.2 presents the assessment meta-model with elements that are specialised for business process capability assessment. The framework was used as base to define the new elements.

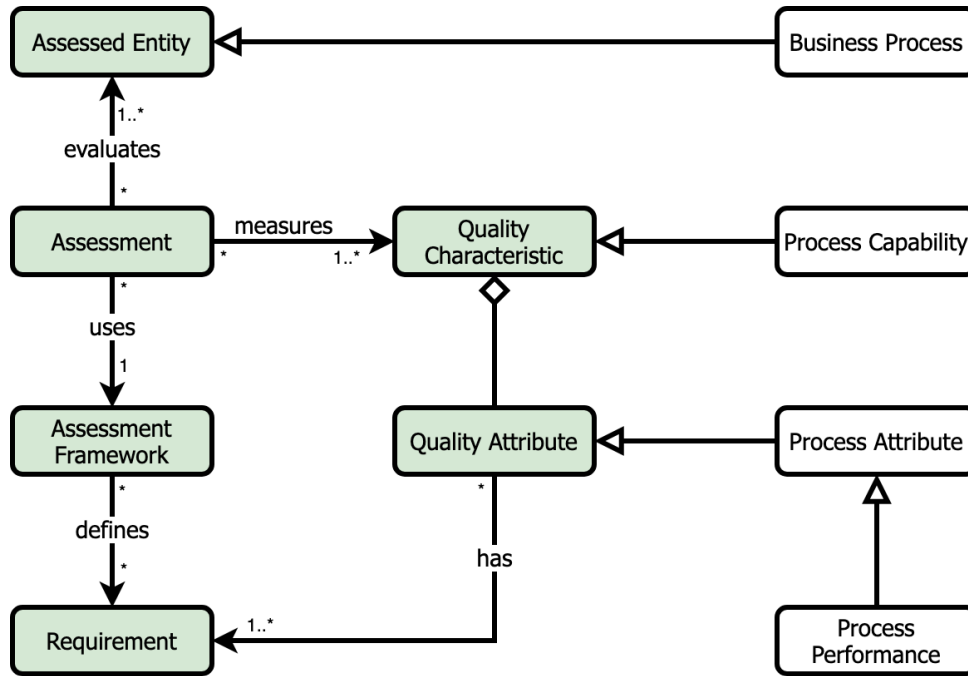


Figure 1.2: A brief overview of the elements of an assessment specialised for business process capability assessment.

The standard defines six capability levels, including the Capability Level 0 defining an *Incomplete process*. The levels are organised in an increasing manner, representing an increasing capability level of the process. Table 1.1 presents the capability levels defined by the standard with their descriptions. The PAs that must be evaluated for each level are also enumerated.

The achievement of a PA is characterised through a rating scale and the capability level of the process is derived using the ratings obtained by each PA. Note also that the achievement of a PA could be connected to the achievement of another PA. ISO/IEC 33020 provides a result calculation method to define the capability level of the assessed process. The method includes a rating scale that defines the degree of achievement of a PA for the assessed process. The scale is ordinal and it is organised as follows:

N - Not Achieved: there is little or no evidence of achievement of the PA.

P - Partially Achieved: there is evidence of some achievement of the PA and an approach to that PA. There is unpredictability of some aspects of the achievement of the PA.

L - Largely Achieved: there is evidence of significant achievement of the PA and a systematic approach to that PA. The assessed process has some weaknesses that are related to the PA.

F - Fully Achieved: there is evidence of full achievement of the PA and a complete and systematic approach to that PA. The assessed process has no significant weaknesses that are related to the PA.

Table 1.1: Process Attributes defined by the ISO/IEC 33020 Measurement Framework grouped by Capability Levels (adapted from [20]).

Level	Description	Process Attributes
1 - Performed	The implemented process achieves its process purpose. The following process attribute demonstrates the achievement of this level.	1.1 - Process performance
2 - Managed	The previously described <i>Performed process</i> is now implemented in a managed fashion (planned, monitored and adjusted) and its work products are appropriately established, controlled and maintained.	2.1 - Performance management 2.2 - Work product management
3 - Established	The previously described <i>Managed process</i> is now implemented using a defined process that is capable of achieving its process outcomes.	3.1 - Process definition 3.2 - Process deployment
4 - Predictable	The previously described <i>Established process</i> now operates predictively within defined limits to achieve its process outcomes. Quantitative management needs are identified, measurement data are collected and analysed to identify assignable causes of variation. Corrective action is taken to address assignable causes of variation.	4.1 - Quantitative analysis 4.2 - Quantitative control
5 - Innovating	The previously described <i>Predictable process</i> is now continually improved to respond to change aligned with organisational goals.	5.1 - Process innovation 5.2 - Process innovation implementation

The rating scale can also be understood in terms of percentage of achievement of a PA as follows: N (0 to $\leq 15\%$ achievement), P ($> 15\%$ to $\leq 50\%$), L ($> 50\%$ to $\leq 85\%$), and F ($> 85\%$ to $\leq 100\%$). The framework also provides a guide to further refine the rating scales P and L using the signs plus (+) and minus (-) to divide them into a total of four ratings: P+, P-, L+ and L-. If this setting is used, the percentage of achievement of a PA for the new ratings are defined as follows: P- ($> 15\%$ to $\leq 32.5\%$), P+ ($> 32.5\%$ to $\leq 50\%$), L- ($> 50\%$ to $\leq 67.5\%$), and L+ ($> 67.5\%$ to $\leq 85\%$).

The framework also proposes three rating methods, which are suggested to be selected by the lead assessor after considering the class, scope and context of the assessment. The rating **method R1** takes into consideration both, the process outcomes and the PA of each process, which must be characterised for each process instance. These characterisations must be aggregated to provide PA and process performance attribute achievement ratings. The **method R2**, on the other hand, considers only the PAs of the processes within the scope of the assessment and not the process outcomes. The PAs must be characterised for each process instance, and the characterisations must be aggregated. Finally, the **method R3** is based on performing the rating of the PAs across the instances of the assessed process without aggregation.

Finally, the framework contains some aggregation methods that can be chosen by the assessment team also considering the class, scope and context of an assessment. The aggregation could be one-dimensional (e.g. when the aggregation is for a PA of one process, or for a process outcome for a given PA across multiple instances) or two-dimensional (e.g. when the aggregation is for a PA for one process, in which the aggregation is done using the ratings of all the process

outcomes for all the process instances). The standard proposes the conversion of the ordinal ratings (N, P, L, F) into interval values (N to 0, P to 1, L to 2, and F to 3; or N to 0; P- to 1, P+ to 2, L- to 3, L+ to 4, and F to 5) that can afterwards be used to apply operations such as the arithmetic mean or the median. The expert judgement of the assessor(s) and the use of certain heuristics are also contemplated by the framework as aggregation methods that can be applied when performing assessments.

1.3 Related Work

Throughout the years, several works aimed at improving the assessment process through the integration of different approaches and technologies into one or more tasks of the assessment [27, 28, 63, 64]. Due to the current trend towards digitalisation in enterprises, recent works are often focused on proposing tools for automating assessment activities. These approaches aim at applying automation techniques to reduce the need for human intervention during an appraisal. Two main lines of research addressed by the scientific community were identified: automated results calculation and presentation phases of the assessment process [27, 28, 30, 55, 61, 89–92], and assessments focused on sensing data from the enterprise to provide assessment results [63, 64, 93].

1.3.1 Automated Results Calculation and Presentation

Regarding improved Results Calculation and Presentation activities of the assessment, the work by [27], which describes the SEAL of Quality Assessment Tool, a software tool for software process assessment developed in the Software Engineering Applications Laboratory (SEAL) of the Electrical Engineering department of the University of Witwatersrand. It aims at supporting assessments compliant with the ISO/IEC 15504 standard [94]. Its main functionality is based on storing the assessment framework in a database. The link associations of evidences identified during the assessment are also able to be stored. Moreover, the tool allows to register the assessment processes, the participants and their roles. It also provides a user interface and the possibility to obtain data visualisations and reports about the assessment. On the other hand, the paper by [28] introduced a knowledge-based decision support system for measuring enterprise performance. The knowledge base contained a set of rules that were used for inference. The data collection mechanism is based on the involvement of top managers that give scores and weights for relative key dimensions based on their own experience. The weights provided by the managers for each dimension are summed and averaged. The input information is used by the system to provide a view of the current state of the performance of the enterprise. The system also includes an artificial neural network in order to forecast future financial measures. Following the same perspective, the work by [89] proposed an ontology-based Records Management (RM) evaluation system. It uses a reasoner that classifies information in a database, containing the baseline and the actual state of the RM system as asserted individuals in the ontology, which was devised by a knowledge-engineer. These individuals are analysed using Semantic Web Rule Language (SWRL) rules defined in the rule base. The final result is calculated by the user interface, which compares the baseline RM system to the current state of the system. To perform data collection, there is a need for an evaluator to input the baseline scenario (considered as the ideal) for the RM system that will be evaluated. For the evaluation, the evaluator must enter the current state of the system. The work by [90] also presented an expert tool for evaluating the Business Intelligence competences of enterprise systems. In the paper, a factor analysis is performed to determine that the intelligence of enterprise systems can be assessed through six main factors,

which are measured through a set of 34 evaluation criteria that are evaluated through a questionnaire. In [61], the authors presented the Software-mediated Process Assessment (SMPA) to automate the assessment of IT Service Management processes. The tool allows to select the process to be assessed and data collection is supported through an online survey. The results are obtained by automatically analysing the collected data to measure the process capability. Improvement recommendations are also provided by the system. The SMPA approach provides four main instruments: assessment survey questionnaire, a method for allocating assessment questions to process roles, the logic to calculate the process capability, and a knowledge base for process improvement compliant to the recommendations defined by the Information Technology Infrastructure Library (ITIL), which is a library that includes a set of best practices to manage and support IT services. The work by [55] presents the AssessAgility software tool that aims at automating and guiding the assessment process based on an exemplar assessment process containing the definitions and guidance to conduct assessments defined by AgilityMod, which is a reference model for performing agility assessment in organisations. The tool provides a graphical user interface that provides distinct sections for each role and responsibility of the assessment process. The data collection depends on the role, for lead assessors, it is possible to register and manage teams, progress of the assessment, assignments, and reports. The assessors, on the other hand, are allowed to register assessments, take notes, and upload evidences. The tool is focused on data collection, whilst the results calculation is performed manually by the assessment team. The work by [30] describes a Software as a Service tool for performing process assessment using the Tudor IT Process Assessment (TIPA) framework. In the tool, data collection is performed manually, and it is user-centred, implying that it is based on the use of Personas, which are representations of real-world individuals in the platform. The latest version of the software (beta) allows to cover almost all activities of an assessment process, from the definition of the assessment to the results presentation activity. All data must be registered manually. In this case, the ratings are manually introduced by the assessors and the final results are calculated automatically by the system. The paper [91] also addressed the TIPA framework by defining an ontology defining TIPA for ITIL concepts, that can be further used to perform inference on the devised knowledge base. Finally, the work by [92] introduced a semi-automated tool for enterprise interoperability assessment that is based on an ontological core to automate the results calculation phase of an assessment of enterprise interoperability.

1.3.2 Automated Data Collection

On the other hand, some approaches introduced ways to connect a system to sensors so as to obtain direct feedback from the environment to be used afterwards for the calculation and presentation of the assessment results, thus allowing to automate the data collection phase of the assessment. The approach by [63], for instance, was devised as the evolution of the CoSEEEK framework introduced by [95], and it focuses on providing automated software engineering process assessment. It is able to support the use of assessment frameworks such as CMMI, ISO/IEC 15504, and ISO 9001. The approach is holistic and ontology-based. The approach supports automatic data collection through the analysis of changes in the artefacts that are used during software development such as source code, test code, reports, and requirements. Results calculation can be performed manually, by allowing human assessors to input ratings to define process capability, and also automatically, in this case carried by a Context Management component that uses ontologies and inference engines to provide the result of the assessment. In [64], an Intelligent Maturity Model Tool was proposed with three main properties: a generic data model enabling the use of different maturity models, it is connected to a Business Process Management

(BPM) system allowing to extract part of the information necessary to perform the assessment and define the maturity level, and an assistant function that recommends improvement suggestions based on the problems identified during the assessment. The data collection method is based on collecting data through e-mail surveys and also providing means to extract data from a BPM system. Finally, the work by [93] presented an approach to perform maturity assessment of information governance. The work is focused on analysing enterprise architecture models represented using ontologies and executing descriptive logic queries to obtain assessment results.

1.3.3 Frameworks to Enhance Assessments

Besides the proposal of automation tools, some works have also aimed at defining frameworks to drive the development of assessment methods, tools, and techniques to improve assessments in enterprises. The work by [96] introduced a framework for assessing the maturity of a system in terms of typical technical performance measures. To validate the framework, the authors provide an illustrative case study based on considering an automobile as a sample complex system to be assessed using the framework. On the other hand, [97] proposed an empirical conceptual framework to assess the awareness of Small and Medium-sized Enterprises (SMEs) regarding the eco-efficiency concept and their capability to implement sustainability strategies. The framework was developed following the DSR methodology and it was validated through an exploratory study carried out on wood industry SMEs. Also regarding SMEs, the work by [98] introduced a framework to assess and manage commercial risk using Artificial Intelligence (AI) methods. The conceptual model, introduced through a workflow, is grounded on the ISO 31000 (2009) standard for Risk Management and it was expanded with AI concepts. The paper also provides recommendations on how to implement the framework. Still regarding SMEs, a conceptual framework for assessing sustainable development in regional SMEs was proposed by [99]. The framework includes external and internal factors that affect the sustainability adoption of SMEs within a regional context. On the other hand, [100] presented an integrated framework for assessing the performance of nonprofit organisations grounded on a literature review. The framework considers seven perspectives of nonprofit performance, and its validity was tested through the development of propositions based on those perspectives in order to explain variation across nonprofits regarding the adoption of different measurement approaches. Finally, the work introduced in [101] proposed the Context-Aware BPM Method Assessment and Selection (CAMAS) Method, which serves to assist method engineers to evaluate the contexts in which their BPM methods can be applied. Moreover, from the perspective of method users, it aids them to select appropriate BPM methods for specific contexts.

1.4 Research Gaps and Contributions

To the best of our knowledge, existing approaches for assessment are focused on tackling specific application domains to evaluate concrete characteristics such as Process Maturity, Enterprise Interoperability, or Enterprise Performance, without following a generic methodology to devise the artefacts they propose. Moreover, most of the frameworks for assessments in organisations are often devised to carry out manual assessments, without introducing proper elements that consider the enhancement of the appraisal process through emerging technologies. Therefore, methods based on the exploitation of capabilities such as automatic reasoning or learning are not considered. Therefore, a generic framework to guide the development of assessment systems that take advantage of novel technologies represents a relevant artefact to improve assessment approaches. This work aims at addressing this gap by proposing a Smart Assessment Framework

(SAF), a conceptual framework grounded on the concepts of smart systems and enterprise assessment to guide the development of automated and semi-automated assessment systems. The SAF is introduced in Chapter 3.

It is worth mentioning that the definition of the elements of the SAF requires the elicitation of systemic elements that allow to take advantage of novel technologies to support assessments. In this context, the elements of smart systems are considered as the core of the framework, since they provide capabilities such as learning, reasoning, sensing, or acting [44], which could provide effectively support the different activities that are performed during an appraisal. In this sense, having a general and clear definition of smart systems is paramount. Therefore, a deep analysis of the literature presenting definitions of smart systems (from different domains) in order to extract common points allowing to define the components of such systems is necessary. Based on the literature review, a meta-model defining the elements of a smart system and the relationships between those elements. Details regarding the literature review and the elements of the meta-model are presented in Chapter 2.

As mentioned before, frameworks focused (*a priori*) on guiding manual assessments are dominant in the literature. This type of evaluation has a number of disadvantages such as the expensiveness of the appraisals in terms of time and resources, and the possibility to obtain incomplete, subjective, biased, or incorrect results. The approaches focused on automating or semi-automating the assessment process are focused on addressing some of these issues. However, these works also present some limitations and drawbacks. For instance, considering the literature presenting tools to automate or semi-automate appraisals, there seems to exist a lack of works exploring the use of unstructured data gathered through emerging technologies to perform assessments. Most works rely on human assessors or experts to define ratings or weights to organisational aspects in order to propagate their decisions through a specific assessment workflow to ultimately generate assessment results. Specifically, text data, which is very common in enterprises in the form of documents, process descriptions, and regulations, is often ignored in this context. Moreover, hybrid approaches combining data-driven and model-driven AI techniques to automate assessments in enterprises are not extended in the literature. This represents an important research gap that is addressed in this work through a hybrid Deep Learning (DL) and Ontology-based approach to automatically treat text evidence, which is introduced in Chapter 4. The approach is an instantiation of the SAF and it uses raw process descriptions in text format as input to perform automatic assessments. The method relies on data-driven AI to obtain partial assessment results from the input evidence and model-driven AI to derive the final results of the appraisal. A software tool implementing the proposed approach is also presented with the defined architecture and its features. It allows users to feed the system with some information and evidences, and to obtain assessment results.

There exists a large number of possible assessment evidences such as text, images, videos, audio, raw sensor data, partial evaluation results, etc. Among them, enterprise models represent an important source of information since they act as structured representations of different organisational aspects such as processes, structure, information, or resources [102]. However, they are generally overlooked by the software systems proposed in the literature. Some research work [103–106] deals with enterprise models for assessment using classical AI methods, often with a combination of knowledge bases and inference rules. However, they are not taking advantage of emerging technologies that are associated to data-driven AI, which have the added value of being able to automatically obtain knowledge from different types of data, without requiring the direct participation of humans to model that knowledge. To address this gap, this thesis presents a DL-based method to treat enterprise models in Chapter 6. Specifically, the approach receives as input business process models and automatically produces preliminary qualitative

ratings that can be afterwards used to derive final results without requiring humans to manually code specific rules or heuristics for processing the models. The DL model is devised to be used as an instantiation of some components of the SAF.

It is worth mentioning that when analysing enterprise models to perform assessments (manually or automatically) there is a need to understand which assessment requirements can be answered with them. In this sense, the modelling language used to design a model has high relevance, since it defines what aspects of the reality can and cannot be represented using its available modelling elements. Some works explore the relationship between enterprise models and the assessment process [107–110]. However, there exists a lack of a methodology to define which requirements defined by an assessment framework can be answered by enterprise models defined with a specific modelling language. The existence of such methodology could allow to reduce the amount of effort required to perform assessments. Therefore, this thesis proposes a contribution in this sense, which is introduced in Chapter 5. Note that the application of this methodology can be considered as a previous step of the development of smart systems in charge of assessments. For instance, a system devised to automatically analyse models defined using ArchiMate and perform assessments based on requirements that are based on the ISO/IEC 33020 framework will be programmed to take into account only those requirements that can be answered through ArchiMate elements, excluding those that are not "visible" in models that are designed using that language. This is not the only possible use that can be given to the approach, other possibilities are explored with more detail in Chapter 5.

Characterising Smart Systems

Contents

2.1	Introduction	13
2.2	Background	15
2.2.1	General Systems Theory	15
2.2.2	The Concept of Smartness in Different Domains	16
2.2.3	Previous Literature Reviews	17
2.3	Systematic Literature Review and Analysis	18
2.3.1	Systematic Literature Review	18
2.3.2	Papers Analysis	20
2.4	Literature Review Results and Discussion	22
2.4.1	Application domains	22
2.4.2	Publication period	25
2.4.3	Text mining results	26
2.4.4	Smartness and Intelligence	30
2.4.5	Similarities within definitions	31
2.5	Characteristics of Smart Systems	32
2.6	Threats to the Validity of the Literature Review	36
2.7	Smart Capabilities in the Context of Design Science	36
2.8	Conclusion	37

2.1 Introduction

The rise of the new industrial revolution, known as the Industry 4.0 [8], is fostering enterprises and governments to adopt and develop new technologies to allow them to optimise strategies, develop new products, shorten development periods, achieve resource efficiency or provide more personalised products. The change from product to service-oriented enterprises is another factor that plays an important role in this trend [8] as well as the extended use of the Internet of Things (IoT) [6]. A relevant aspect in this new scenario is the special focus on the reduction of the breach between physical and cyber components, accomplished through the integration of operation systems and Information and Communication Technologies (ICT) with the idea of forming Cyber-Physical Systems (CPS) [111]. Moreover, there is a trend based on moving

from the development and application of computer-aided approaches [112–114] to more digital, networked and intelligent methods and agents [115,116]. In this context, the concept of Smartness and Smart Systems have begun to be used in different domains. Indeed, the word "smart" has been widely used to name products (Smart Phone, Smart TV, Smart Watch, etc.) as well as methods, applications, frameworks, and techniques proposed in the scientific literature [117–121]. Recently, concepts such as Smart Cities [122], Smart Manufacturing [123], or Smart Grids [124] have gained attention not only being tackled by industry practitioners but also being widely discussed in the scientific literature. Generally, these concepts are associated with advances within specific fields through the development, implementation and application of ICT [125,126], the use of intelligent devices [127], Artificial Intelligence (AI) [128], and novel industrial and management technologies [126].

Notwithstanding, this wide adoption of the word "smart" to represent or qualify certain types of systems did not provide a unified specification of the features that make a system smart. The scientific literature shows divergences when describing Smartness or Smart Systems. Some works consider that smartness is possible through the extensive use of ICT or digital capabilities [124,125,129–133], whilst others provide a more holistic view of the concept, considering that smartness is reached through the cooperation of several factors such as policies, economy, governance, education, individuals, technology, sustainability, etc., towards the improvement of a certain environment and its inhabitants [134–136]. In the business and management field, smartness is defined by the business strategy, the background of the stakeholders or behaviours directed towards sales [137–139]. On the other hand, in the field of social sciences and education, smartness is viewed as a social construct instead of the inherent and interrelated abilities of an individual [140,141]. In this sense, there is a need for reaching a common understanding of the concepts of Smart Systems and Smartness to guide the development of novel methods and technologies. However, although an understanding may have been reached in some domains and, in some cases, more general concepts are introduced [142,143], there is a lack of a generic view of the concept of Smart System that integrates different domains into a unified concept.

The exploration of the state of research regarding Smart Systems and the concept of Smartness within certain domains is an open research subject. In general, works addressing this subject aim to obtain definitions to serve as common ground for the development of new applications. Notwithstanding, most papers do not consider a generic view that is independent from the application domain. Moreover, there seems to exist a lack of consensus regarding the conceptualisation of Smart Systems and their characteristics considering common points regarding how they are treated in different domains in the scientific literature. Therefore, the work presented in this chapter aims at exploring the state of the art regarding Smartness and Smart Systems. A special focus is put on defining the characteristics of a Smart System through the study of explicit definitions of Smartness and Smart Systems presented in scientific papers. This is done by manually extracting those definitions and analysing them through diverse methods.

To this end, a Systematic Literature Review (SLR) aiming to retrieve papers containing definitions of Smartness or Smart Systems is conducted. The search is performed with independence from the domain in which the definitions are presented. The definitions are then manually extracted from the collected papers and a semi-automatic analysis is carried out to obtain the results. The final objective of the work is to identify the characteristics of Smart Systems, driven by similarities between the concepts presented within the extracted definitions. The work aims at answering the following research question:

- What are the characteristics of Smart Systems?

The contributions of this chapter can be summarised as follows: (1) to provide a view on the

state of the art regarding of Smartness and Smart Systems considering how they are defined, (2) to identify similarities in the definitions provided in scientific papers regardless of the application domain, (3) to define the characteristics of Smart Systems, (4) to establish an initial common ground towards a unified definition of Smart Systems through a formalisation represented using a meta-model, and (5) to provide conceptualisations capable of serving as part of a knowledge base to be used for the development of the Smart Assessment Framework (SAF).

This chapter is organised as follows. Section 2.2 introduces concepts that are relevant to describe the work and it describes the attempts of defining different types of smart systems in the literature through literature reviews. Section 2.3 describes the methodology that is followed to perform the SLR and the approaches that are used to analyse the selected papers. Section 2.4 describes the results obtained from the analysis. Section 2.5 presents the characteristics of Smart Systems considering the obtained results. Section 2.6 enumerates the threats to validity of this work and the manner that those threats are addressed. Section 2.7 describes how the work presented in this chapter fit in the context of the Design Science Research (DSR) methodology. Finally, Section 2.8 presents the conclusions and future research perspectives.

2.2 Background

To define the characteristics of Smart Systems, it is worth to first explore the definition and properties of a system from a generic perspective. Therefore, in this section, some systemic notions defined in the General Systems Theory [78] are described. After, an introduction of different domains in which the concept of Smartness is used is presented. Finally, the works that aimed at exploring the state of the art regarding different types of Smart Systems such as Smart Cities, Smart Homes, Smart Industries, etc., are enumerated and described.

2.2.1 General Systems Theory

The most accepted definition of a system was introduced by [78]: "*a system is a set of interacting elements*". [144] extended this definition considering that, according to GTS, a system is a set of interconnected parts with properties that are richer than the sum of the properties of those parts. Moreover, a system has boundaries and a set of attributes that make it unique when compared to other systems [144].

There are two fundamental perspectives to view a system [145]: from the *teleological* aspect and from the *ontological* aspect. The *teleological* aspect is focused on the behaviour and functioning of the system, whilst in the *ontological* view the system is analysed from a structural perspective.

The work by [144] based its definition on the *ontological* view by considering a system as a "*bounded set of inter-connected elements forming a whole that functions for a specific finality in an environment, from which it is dissociable and with which it exchanges through interfaces*". The authors also defined a list of concepts that characterise a system:

- System Element: a component of a system.
- Relation: a link between two entities.
- Objective: set of actions that the system can execute in its environment in order to achieve its objective.
- Environment: all that is outside the boundaries of the system.

- Interface: element of the system through which a connection between the system and its environment is established.

In this work, these are considered as base characteristics. Likewise, a smart system has all of those base characteristics and some additional ones that this work intends to define.

2.2.2 The Concept of Smartness in Different Domains

The Smartness concept is used in a variety of domains, representing different types of systems. This section briefly describes the main concepts behind some common Smart Systems next.

Smart City

It is based on the idea of creating an urban space characterised by sustainable economic growth and high quality of life of its inhabitants [146]. The core of Smart Cities is the interconnection between all possible elements of the city, including citizen and business, and interfacing them one to another through state-of-the-art technologies [147] such as integrated materials, sensors, computer networks, and decision-making algorithms [148]. Smart Cities include different application areas such as Governance, Energy, Building, Mobility, among others [149], and there are currently various advances associated to the concept such as 5G technologies, sensors, IoT, robotics, virtual and augmented reality, and artificial intelligence [147].

Smart Home

Smart Homes are based on the implementation of a communication network that allows to connect and remotely control or monitor devices and services [150]. The devices used for Smart Homes are various including air conditioning and heating, ventilation, lightning, hardware, and security systems [151]. These devices are often named "gateways", and they are able to interact with tablets, mobile phones, etc [151]. The objective behind the concept is to improve the lives of people through socially and time assistance, guided by automated activities such as sensing, anticipating, and responding to events within a house [152].

Smart Industry

To understand Smart Industry, it is first necessary to define the ideas behind Smart Factories, which are a flexible cyber-physical production factories that provide personalised services or products to customers and their functioning is based on the use of Big Data [153]. In this context, Smart Industry is an evolving network of Smart Factories that fosters and is influenced by the participation of researchers, developers, suppliers, distributors, and end users [153]. Moreover, the Smart Industry is possible through a synergy of IoT devices that allow to enhance the use of computers in the industry [154]. A common term for the Smart Industry is the "Industry 4.0" [155], which was introduced by the German government, and it is based on the IT-driven transformation of manufacturing systems [8] grounded in concepts such as CPS [4], IoT [6], decentralised decision-making, personalised products and services, etc. In this work, the terms "Smart Industry", "Smart Manufacturing" and "Industry 4.0" are used interchangeably [2].

Smart Service System

Service systems are a systems composed of people, information, organisations, and technologies that interact towards the achievement of common objectives [156]. Service systems depend

not only on the system elements by themselves but also on the manner that those elements interact [157]. On the other hand, Smart Service Systems are service systems able to learn, perform dynamic adaptation, and make decisions considering data that is received, transmitted or processed in order to improve their capabilities in the future [158]. This type of service system put special emphasis on the human, so that aspects such as knowledge, capabilities, and value are defined by the people of the system [159].

2.2.3 Previous Literature Reviews

Several works performed literature reviews during the recent years with the objective to provide a view of the state of the art regarding different Smart Systems such as Smart Cities, Smart Industries, Smart Homes, and Smart Service Systems. This section briefly describes those works, grouping them according to specific application fields for better organisation and to allow the readers to easily highlight how the vision of smartness varies between fields.

Regarding **Smart Cities**, the work by [160] carried out a literature review with the objective to explore definitions, dimensions, models, and paradigms around the domain. On the other hand, the work by [122] aimed at obtaining the definition of Smart Planning considering its spatial dimension by exploring the use of the word "smart" in the scientific literature. The authors defined the pillars of the procedure of Smart Planning: intelligent location, built environment, smart energy management, and smart mobility for smart people, putting special emphasis on the role of people and their adaptation to smart solutions to tackle the problems within a city. The authors claim that a smart city as a concept involves the urban development in a cohesive strategy through the combination of technology and the knowledge and role of the ICT.

As for **Smart Homes**, [161] presented a literature review describing the goals of Smart Home Energy Management System and its definitions, applications, and how the components of a Smart Home are manufactured. A literature review is also presented in [162], the work focuses on home automation systems considering their features, benefits and drawbacks. Moreover, [163] analysed 48 papers on smart homes and home health monitoring technologies for older adults. Conclusions drawn from the analysis point out that the technology readiness level for Smart Homes and Home Health Monitoring was still low at the time of the publication. [151] also presented a systematic literature review addressing Smart Home applications related to IoT. The work analysed 229 papers and enumerated the motivations of the use of Smart Home applications based on IoT, the challenges that may emerge, and the recommendations to address those challenges.

Concerning **Smart Manufacturing** or Smart Industry, despite that this naming is still used in the literature, the Industry 4.0 is often considered as equivalent [2]. In this domain, [123] published a literature review based on an analysis of national strategies and policies addressing Smart Manufacturing from various countries as well as the available scientific literature introducing relevant topics. The work concludes that the most important issues for achieving Smart Manufacturing are the interoperability and development of technologies. [155] also discusses the past, present and future of Industry 4.0 through a Systematic Literature Review (SLR) method that aimed at characterising the state of research for the domain. The work by [164] followed the same method and discussed the current state of research regarding Industry 4.0 including key technologies and applications. Moreover, [2] followed the same path and performed a bibliometric analysis in addition, aiming to describe a detailed overview on the domain including related concepts such as IoT, Industrial Wireless Networks, Cloud Computing, Energy-based Systems, among others.

Some works addressing **Smart Service Systems** have also focused on the search for a conceptualisation through approaches based on performing literature reviews. The work presented

in [149], for instance, aimed at defining Smart Service Systems by applying metrics and machine learning algorithms to perform text mining on a large corpus of scientific papers and news articles. The study puts special focus on Smart Service Systems applied to different aspects of human life such as homes, energy, healthcare, and transportation, and it provides relevant keywords, research topics, technology factors, and application areas regarding the studied subject. The work by [165] also aimed at conceptualising Smart Service Systems through a literature review and in-depth interviews with informants from the industry that have begun introducing smart products into their service systems to improve the capabilities of those systems.

The works presented in this section show a tendency to perform domain-specific studies. Moreover, most works are not focused on obtaining characteristics that make systems smart. This work aims at closing these gaps by performing a domain-independent search of definitions of Smartness and Smart Systems in scientific papers and analysing those definitions to extract the capabilities of smart systems.

2.3 Systematic Literature Review and Analysis

There are two main activities in the literature review presented in this chapter. Those activities are shown in Figure 2.1 through a Business Process Model and Notation (BPMN) diagram [166]. The first activity is based on retrieving a set of papers containing definitions of Smartness and Smart Systems in the scientific literature. This activity produces a set of selected papers that are then used as input to the next activity, which is based on analysing the definitions contained in the papers through text mining methods.

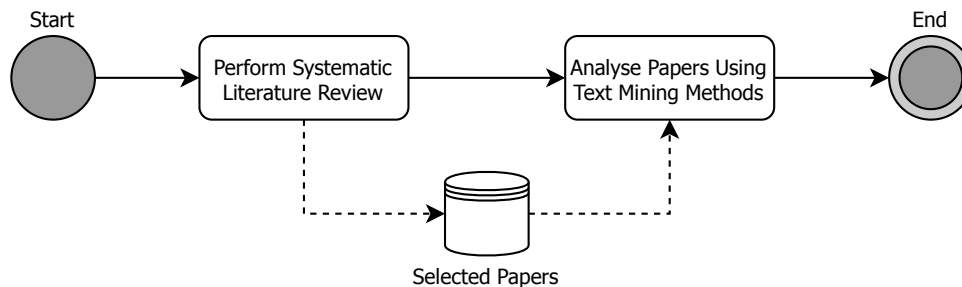


Figure 2.1: Overview of the methodology followed in this work.

2.3.1 Systematic Literature Review

An SLR is based on the application of a systematic process to define the research question, identify relevant studies, evaluate their quality and summarise the findings qualitatively or quantitatively [167]. Moreover, the tools used for selecting the studies must also be identified [168]. The search performed in this work is based on the methodology introduced by [169]. In general, systematic literature reviews are based on carrying out a cycle composed of three main phases: (1) define keywords, (2) search in the literature, (3) analyse the results [170]. The main advantage of performing an SLR is that it follows a predefined strategy that allows to provide concrete evidence regarding the data sources and the criteria to select and analyse papers [77].

The SLR carried out in this work is a process that starts with the definition of the research question and is followed by the definition of the keywords, the search strings and the databases.

The search is then performed by querying the databases using the search strings. A filtering sub-process is then carried out, and it is based on analysing metadata from each paper to define if it satisfies the first set of criteria F_1 . The full text of a paper that satisfies F_1 is then examined in order to check the second set of inclusion criteria F_2 . The paper is included if F_2 is satisfied. Any other paper is discarded. Details regarding each activity of the methodology are described in the following sections.

Research question

Initially, the research question is defined. Note that the objective is to obtain a characterisation that is independent from the application domain. Hence, the research question that drives the SLR is:

- RQ: What are the characteristics of Smart Systems?

Considering RQ, the following specific questions are derived to serve as guide for the analysis of the papers gathered with the SLR.

- What are the domains of the papers containing definitions?
- What is the time period in which the papers were proposed?
- What are the main keywords used by the authors to propose their definitions?
- Are there any common aspects in the definitions when considering different application domains?
- Considering common points, what are the characteristics of a smart system?

Keywords and search strings definition

The keywords used in the search strings are *smartness* and *smart*. Two search strings are defined, both focused on obtaining explicit definitions, considering that explicit conceptualisations are most likely to include the characteristics of the system that is being defined:

- S_1 : $\{smartness\ is\}$, focused on the first keyword *smartness*.
- S_2 : $\{we\ define\ smart^*\}\ OR\ \{smart^*\ is\ defined\}$, related to the second keyword *smart*.

Note that S_2 is very specific, whilst S_1 is open to diverse string combinations. This structure for S_1 is selected since the concept of Smartness may not be explicitly defined, but described as an attribute of a system instead.

Database Selection

Since two separate search strings are defined (S_1 and S_2), and one is much more specific than the other, two different searches are carried. For S_1 , the following databases are used: ACM, Scopus, IEEEExplore, Taylor & Francis Online, and Web of Science. Five databases are selected taking into account that a minimum of four is sufficient to perform a robust literature search [169]. Since S_2 is much more specific than S_1 , the number of results obtained with this search string is expected to be much lower. Hence, the databases SpringerLink and Wiley are added to those used for S_1 .

It is worth mentioning that Scopus and Web of Science are used, which are extensive abstract and citation databases that gather papers from several peer-reviewed journals. The papers that can be retrieved with these databases come from diverse publishers such as Elsevier, Springer, Taylor & Francis Online, and IEEE. It is expected that this will provide more robustness to the search. Naturally, duplicated papers are expected to appear and they are removed. Each database has its own syntax to write queries. Hence, the search strings presented in the previous section are slightly modified for each database to obtain the expected outputs.

Papers Filtering

After defining the research question and the databases to perform the search, the first filter F_1 consists of basic inclusion criteria comprising: papers in English, papers from journals or conferences proceedings, papers with available full texts, and primary studies. The second filter F_2 consists of the following mechanism: for each retrieved paper we perform an initial exploration to define if it contains definitions for Smartness or Smart Systems. This analysis consists of searching the words "smartness" and "smart" within the text and analyse the context of the paragraph in which the word is present to define if it is a definition. If no definition is found, an overview of the title, abstract, introduction and conclusion is performed in order to identify possible implicit definitions. Papers containing definitions are selected and the proper conceptualisations are extracted, note that there may be more than one definition per paper, hence the number of definitions may not be equal to the number of selected papers. Table 2.1 summarises the inclusion criteria and presents the exclusion criteria derived from them.

Table 2.1: Inclusion and exclusion criteria used to select the papers. The criteria are organised in two filters, F_1 and F_2 , the first is related to metadata analysis whilst the second is based on analysing the full text of the papers.

Filter	Inclusion Criteria	Exclusion Criteria
F_1	Papers written in English Paper with full text access Primary studies	Papers not written in English Paper without full text access Literature reviews
F_2	Papers containing definitions Papers with implicit definitions	Papers without definitions Papers without implicit definitions

2.3.2 Papers Analysis

After extracting the definitions from the selected papers, text mining techniques are applied to perform a detailed analysis. This allows to provide a quantitative, and therefore objective, vision of the set of definitions. The activities that compose the methodology followed in this work are presented in Figure 2.2 through a BPMN diagram.

Note that three parallel paths are introduced in Figure 2.2. Each path represents one approach to analyse the definitions that were collected. The common activity to all paths is the extraction of definitions from the selected papers obtained after applying the methodology described in Section 2.3.1.

The first path (bottom path in Figure 2.2) is based on manually extracting keywords from the definitions and calculating their frequency. The output of the final activity of the path is a

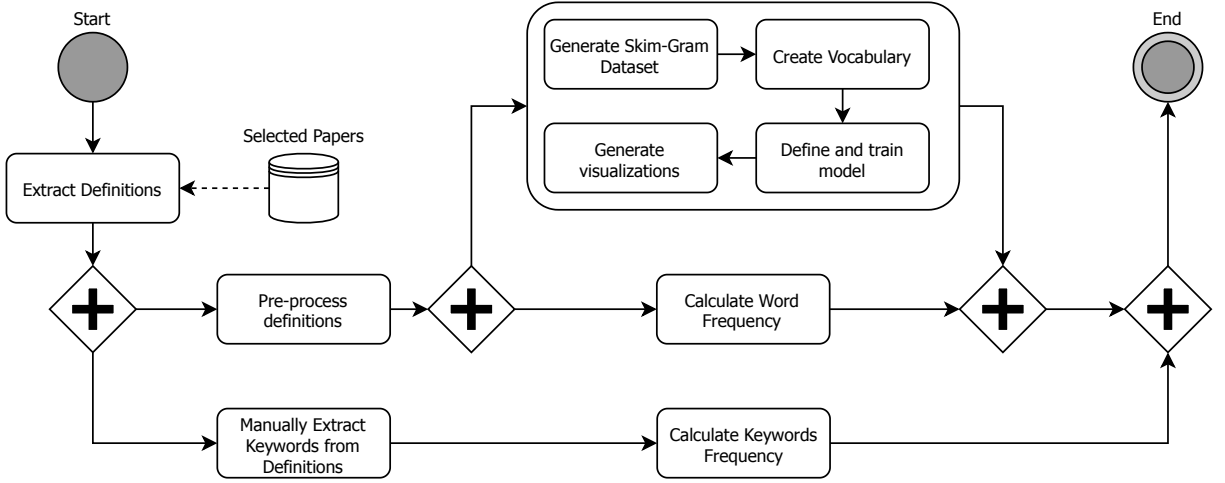


Figure 2.2: Overview of the methodology to explore the definitions through text mining techniques.

set of T tuples $(w, f)_i$ where $i \in T$, and w and f are the i -th word and its frequency, respectively.

The second path is composed of two other parallel paths. For both sub-paths, an activity based on pre-processing the definitions is carried out. This pre-processing step consists of removing punctuation signs, such as dots, commas, etc., and removing stop-words. A total of 179 words that are considered as stop-words are removed from the text, some of them are *his, any, down, you, can, against*, among others. Finally, stemming is also performed in order to convert every word into its root version, e.g. *intelligence* and *intelligent*, are converted to the root word *intellig*. Applying stemming is specially relevant for document retrieval algorithms due to that, depending on the chosen stemming method, it may allow to improve Recall without a significant lost of Precision when querying documents [171].

For the first sub-path (middle path in Figure 2.2), the frequency is calculated as described for the first main path. However, the calculation is applied on the full texts of the definitions and not only on the manually extracted keywords. The second sub-path is composed of a sub-activity based on the construction and training of a Deep Learning (DL) model [13]. The method used in this work, based on using learned representation of words with vectors, is currently widely used in for Natural Language Processing (NLP) [13]. The core of the method is the Skip-gram model [172], which is a neural network devised to learn vector representations of words within a text corpus. The final idea is to use this vector representation to graphically visualise the relationship between the words within the definitions. For this purpose, the t-Distributed Stochastic Neighbor Embedding (t-SNE) [173] is used to reduce the dimensionality of the vectors; and the K-Means [174–177] algorithm is applied to organise the vectors into ideally representative groups for pattern identification. K-Means was selected due to factors such as ease of implementation, simplicity, efficiency, and previously proven empirical success [178].

This sub-activity starts by concatenating all pre-processed definitions into a single string s , which do not have punctuation and stop-words, and it is composed only of root words. In the following phase, a vocabulary used to numerically represent the words of s is created. This consists of assigning an integer value to each word of s . Next, a dataset containing M tuples $(l, t)_j$ composed of indices from the vocabulary is generated, where $j \in M$, and each tuple j consists of an input word l_j and a target word t_j that appears close to l_j in s within a certain

window size C . Hence, M ultimately depends on the size of C , the largest the window size, the more tuples are generated. The following step is based on defining and training a Skip-gram model using the dataset of tuples. Skip-gram is a neural network [179] that receives as input \vec{x} encoded as a one-hot vector. The output of the network is a vector \vec{y} , with the same size as \vec{x} , that contains at each of its positions the probability of the word l_j to be close to one word of the vocabulary. In other words, the model is trained to output highest probabilities for words that are more likely to appear next to the input word. For this purpose, the tuples $(l, y)_j$, previously encoded as one-hot vectors, are used as input and target output of the network respectively during training.

The idea of training Skip-gram is to obtain vectors, which are the weights of the network and also known as word embeddings, that represent each word considering the specific context of the word within the text that was used to train the model. This may allow to perform different tasks such as sentiment analysis, text querying, and word visualisation, which is the reason why the method is applied in this work. More details regarding the model can be found in the original paper [172].

2.4 Literature Review Results and Discussion

The method described in Section 2.3.1 allowed to obtain a total of 123 papers containing a total of 177 definitions. The number of papers obtained for each database is presented in Table 2.2 joining the results for S_1 and S_2 .

Table 2.2: Number of papers retrieved from each database, papers containing definitions, and definitions extracted from the papers.

	Retrieved	With Definitions	Definitions
ACM	16	5	9
Scopus	33	14	17
IEEEExplore	167	58	86
Taylor & Francis	74	17	22
Web of Science	19	5	8
SpringerLink	40	21	29
Wiley	8	3	6
Total	357	123	177

The definitions were various and most of them were introduced in their respective papers targeting specific application domains. Table 2.3 shows some examples of the papers, including the manually extracted definitions, keywords, and domain. Note that, for some cases, more than one definition is introduced in a paper. An example of this scenario is provided in the table with the work by [180], which provides four definitions of Smartness.

2.4.1 Application domains

The retrieved papers were part of various domains since the search string was not restricted to specific applications. Figure 2.3 presents the domains addressed by the papers. The domains were manually defined after analysing the content of each publication. The papers were organised

Table 2.3: Examples of gathered papers with the extracted definitions, keywords and domain.

Paper	Definition	Keywords	Domain
[181]	Smartness is attributed to the development and utilisation of knowledge, skills and behaviors to resist educational hegemony in a way that is beneficial to self and the larger community.	Development, knowledge, skills, behaviors, benefit, self, community	Education
[182]	We define smart cities, which are intelligent large-scale environments, with large amounts of data that are used to improve the quality of life for its citizens.	Intelligence, large-scale environment, big data, improve, quality of life	Infrastructure
[183]	We define smart to mean embedded actuation, sensing, and control logic in a tightly coupled feedback loop.	Embedded, actuator, sensor, control, feedback	Business and management
[136]	"Smart" is defined by the deployment of innovative and commercially available state-of-the-art technology.	Innovate, commerce, technology	Manufacturing
[184]	We define smartness as the number of nodes that have been connected to the information network. Therefore, smartness conceptually represents the degree to which a particular smart grid technology and its external links has been integrated into the utility's power network.	Node, connected, information, network, grid, integration	Energy
[185]	The smartness is coming from the algorithms that extract unique features and recognise the patterns in the features.	Algorithm, features, pattern recognition	Chemistry
[186]	Smartness is associated to the system capability of self-adapting when the operating conditions change and usually it resorts to the action of the control system.	Self-adaptation, action, control	Electronics
[187]	We define smart spaces as environments in which objects have associated with them: (1) a CPU (or brain) allowing some minimal level of intelligence to exist in the object, and (2) sensors that detect the presence of other objects within the environment or the state of the object itself.	Environment, CPU, brain, intelligence, sensor, detect	Theory
[180]	Smartness is being connected and able to act with partial autonomy based on own decision making while cooperating and coordinating with others. This "smartness" typically stems from highly cooperative behavior, self-awareness, self-adaptation, and self-optimisation. Smartness allows the system to introspect and reason about its environment (including potential security threats), the system can protect itself. For a Smart Cyber-Physical System smartness is the system's ability to learn and respond intelligently in new contexts.	Connected, act, autonomy, decision-making, cooperation, coordination Cooperate, self-awareness, self-adaptation, self-optimisation Introspect, reason, environment, self-protection Learn, respond, intelligent, context	Information Technology

so as to be part of exclusively one domain. The domain "Theory" comprises theoretical works and papers proposing conceptual models.

As presented in Figure 2.3, the concept of Smartness is addressed in various domains, such as

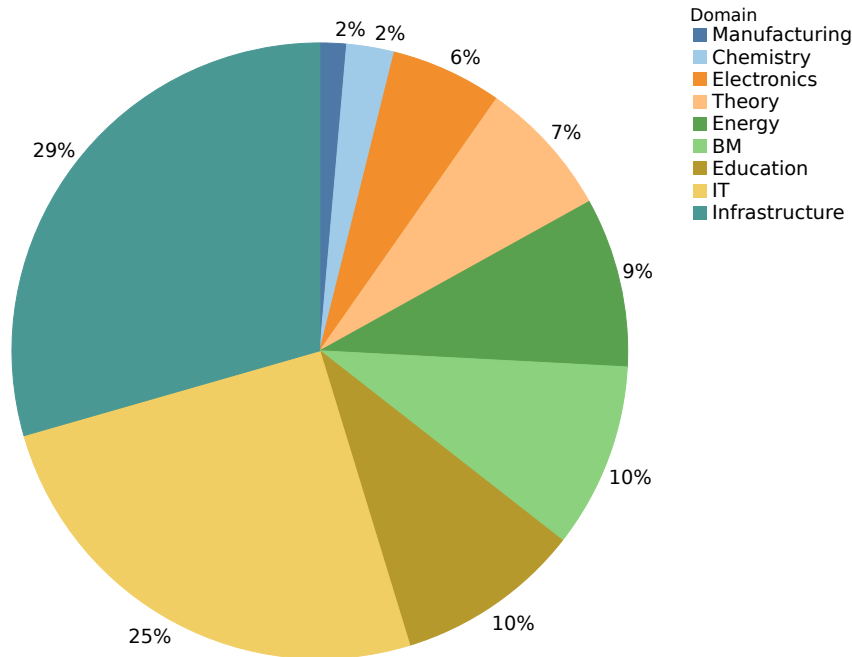


Figure 2.3: Domains of the papers that presented definitions of Smartness or Smart Systems.

energy, manufacturing, education, among others. The most addressed domain is *Infrastructure*, with almost 30% of the papers. The domain includes papers regarding smart buildings, cities and infrastructure. This result is expected due to the spread of the use of the term "smart" within this domain for different purposes. On the other hand, the *Information Technology (IT)* domain groups 25% of the papers. It includes all sub-fields related to the treatment of information such as Information Architecture, Information Retrieval, Communication Technologies, among others. We also included papers related to Computer Science within this domain. Together, these two domains (IT and Infrastructure) comprise more than half of the papers, showing that there is a strong tendency of scientific publications of these domains to introduce definitions.

It is worth mentioning that the domain of each paper was considered when extracting the keywords presented in Section 2.3.2. The purpose was to separate aspects that were domain-specific so as to obtain keywords that could be applicable for Smart Systems from a generic perspective. However, this was not always possible since some characteristics mentioned in the definitions were too specific for the domain. In such cases, the keywords were discarded.

As shown in Figure 2.3, the concept of Smartness is not employed for a particular domain, but they have spread to different applications. In some cases, the definitions considerably differ one from another, but there are interception points that are common for most applications.

The common aspect for most of the definitions found within these domains is that technology is used to provide smart capabilities to some system. However, the literature review also revealed some definitions that are not directly related to the use of technology to develop Smart Systems. Nevertheless, they provide relevant insights towards the achievement of a generic understanding of Smartness. For instance, considering the domain of *Business and Management*, [137] define smart capital as the flow of information from a company to the financier, in which there is

consultation and support that flow in the opposite direction. The authors also state that the degree of smartness depends on the financial product used, the institutional background and business strategy of the financiers, expected time-horizon of the investment, and the stage of development of the firm that is being financed. In the same domain, the concept of working smart is also present in the literature, the works by [138] and [188] present definitions initially introduced by [189]. The former presents the explicit definition introduced in the original paper which states that working smart comprises "behaviors directed toward developing knowledge about sales situations and utilising this knowledge in sales situations", whilst the later also relates the concept with the optimal behavior in sales situations that is based on making situation-appropriate decisions. These concepts are directed towards the use of data or knowledge to make decisions for business purposes.

As for the definitions that consider social aspects, the work by [141] refers to smartness as a socially constructed concept that divides people into two groups (the smart ones and the ones that are not). The idea that smartness is a concept created by the society is also mentioned by [140], who states that it is a cultural practice rather than an inherent biological property of an individual, criticising the tendency of schools to overlook the knowledge that comes from personal experiences of the students. Other authors present smartness as a relationship between an individual and its community. [181] states that smartness implies the development and use of knowledge, skills and behaviours in order to obtain benefits for the individual and its community by resisting the educational hegemony, and [190] relates smartness to the balance between knowledge and wisdom with strong commitment to the community and work ethic. Some authors relate smartness with the possession of certain abilities that allow smart individuals to be labelled as such. [191], for instance, defines smartness as an understanding of the world with the ability to act with a purpose, think rationally, and effectively dealing with the environment. The definition of [192] also mentions the ability to interact with the world with low effort to produce some outcome. A similar concept is presented by [193], who defines smartness as being capable to respond to some feedback with a quick mind. [194] makes the distinction between book smartness and street smartness, presenting the former as explicit knowledge and the later as the tacit understanding capability of an individual. Finally, [195] defines smartness in the context of smart power as the deliberation, self-knowledge, self-control, creative imagination, and knowledge about the world used in combination to foresee possible consequences; this ability allows to take strategies that are specifically designed and adapted to a particular context even when there are already available some pre-defined "standard procedures".

Other definitions include smart data, defined by [196] as significant and semantically explicit data that is useful to fulfil objectives; smart office, defined as aspects of business processes that coordinate the operations in the office within the context of projects and tasks that are performed by employees on a regular basis [197]; smart service systems, which are systems that rely on smart products as objects that interface service consumers with service providers to coordinate their resources and activities [165]; smart packaging, which employs systems to perform automatic and autonomous modulations in micro/nanostructures in order to adapt to dynamic changes in the environment [198].

2.4.2 Publication period

Considering the time of publication, Figure 2.4 shows that there is a tendency to provide or explore definitions of Smartness or Smart Systems only during the recent years. This could be due to a possible augmentation of the use of those concepts in both the industry and academia.

Figure 2.5 presents the number of papers per year considering the application domain. Results

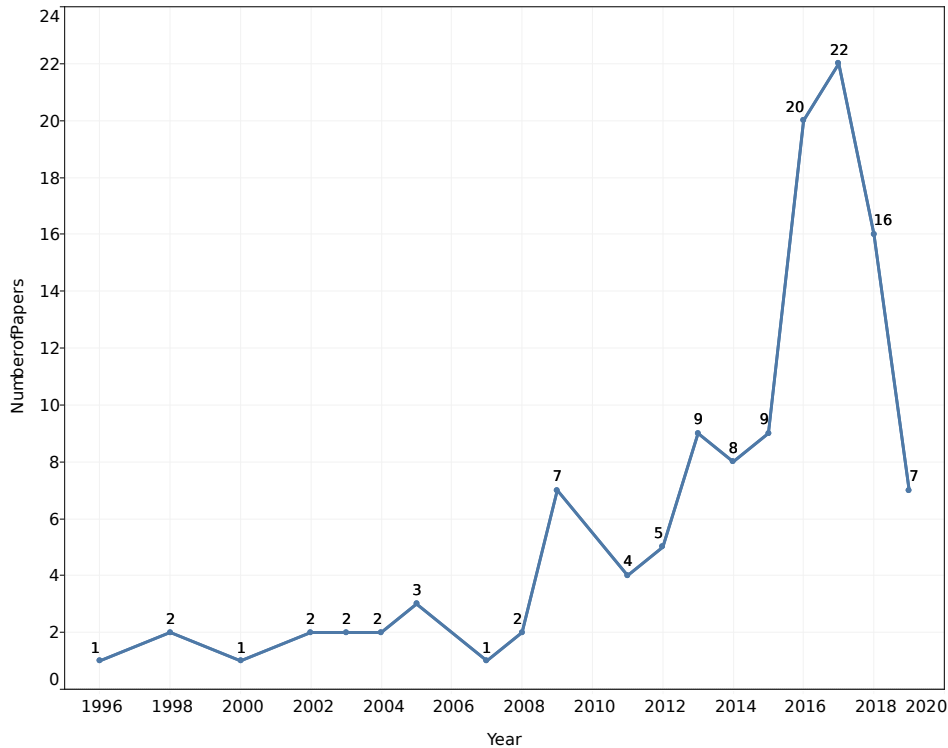


Figure 2.4: Number of papers containing definitions per year.

show that there are papers presenting definitions since the late 90s until the recent years for the domains: *Information Technology*, *Theory*, and *Business and Management*. There seems to be an interest of exposing the concept of Smartness, specially in the *Information Technology* domain, which is the domain in which the number of papers is the most distributed throughout the years. Although *Infrastructure* is the domain with the largest number of papers, the discussion of the Smartness concept has been addressed the most during the last years starting from 2015. However, the number of papers introducing the concept of Smartness within the *Manufacturing* domain is very low, this may be caused by the tendency to use and explore the term Industry 4.0 instead of Smart Manufacturing or Smart Factory.

2.4.3 Text mining results

This section presents the results of the methodology described in Section 2.3.2. First, the results of the word frequency method considering the full definitions and the extracted keywords are presented and discussed. Next, details of the implementation of the Skip-gram model to analyse the full definitions are described, and the visualisation of the most frequent words using the word embedding vectors is shown and analysed. After, the different views that were found in the literature regarding the relationship between smartness and intelligence are described. Finally, a final analysis of the common aspects of the conceptualisations of Smartness and Smart Systems considering the different domains is presented.

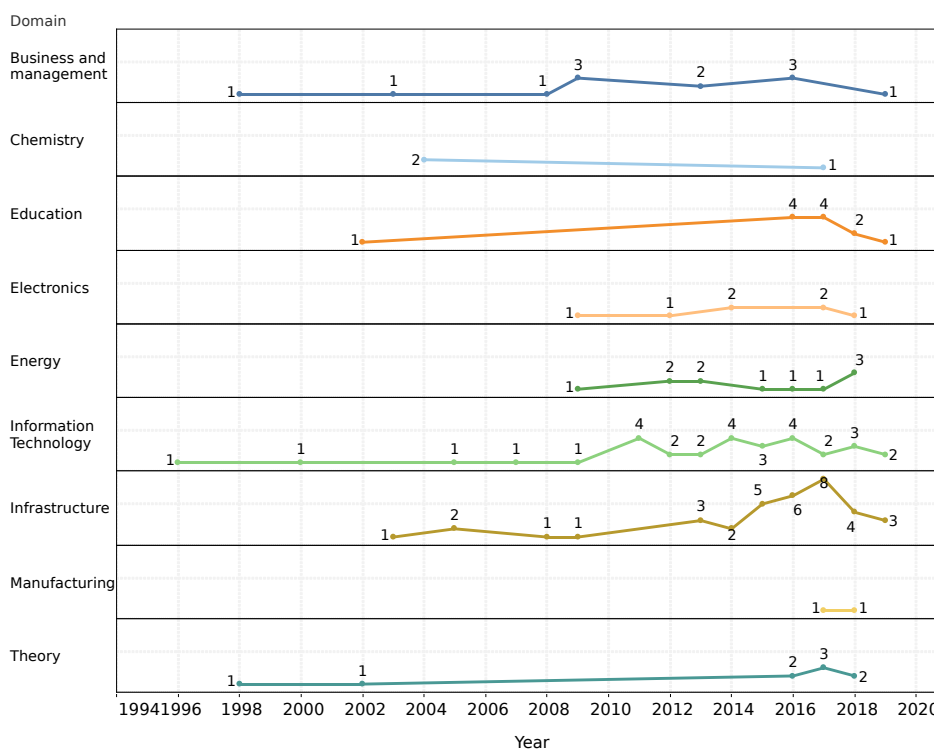


Figure 2.5: Number of the papers that presented definitions of Smartness or some Smart System per year considering the domain.

Word frequency analysis

Word frequency counting was applied to the full definitions of the papers. The frequencies of the top-25 most frequent words are presented in Figure 2.6. Some considerations regarding the figure are worth mentioning. Only the root of the words (stemming) are considered. The word "smart" is by far the most frequent one with 128 appearances so it was not included for the analysis. The roots "defin" (root of words such as "define", "defined", "definition", etc.) and "use" were also removed. The first one is very frequent due to the search string S_2 and it does not add much information regarding the investigated subject. As for the word "use", it is too generic to provide relevant insights.

The most frequent word in the definitions is "system" with 41 appearances. The fact that "technology" is in the second place complies with the idea presented by some authors who state that smartness is possible through the use of ICT. Moreover, other frequent words such as "sensor", "devic", "physic", "object", "control", "environ" and "interact" could imply a tendency to use technology considering the following scenario: devices that sense, control and act on objects that are present in an environment in the physical world.

It is also relevant to point out the frequency of the words "user" and "human", implying that Smart Systems ultimately aim at serving users. The treatment of data ("data" and "inform") and its use for learning ("knowledg", "learn") seems relevant as well in the definitions along with the communication capability ("commun", "interact", "integr", "network"). On the other hand, the word "intellig" is very frequent. Therefore, there seems to exist a connection between smartness and intelligence. In fact, this relationship is explored in detail in Section 2.4.4.

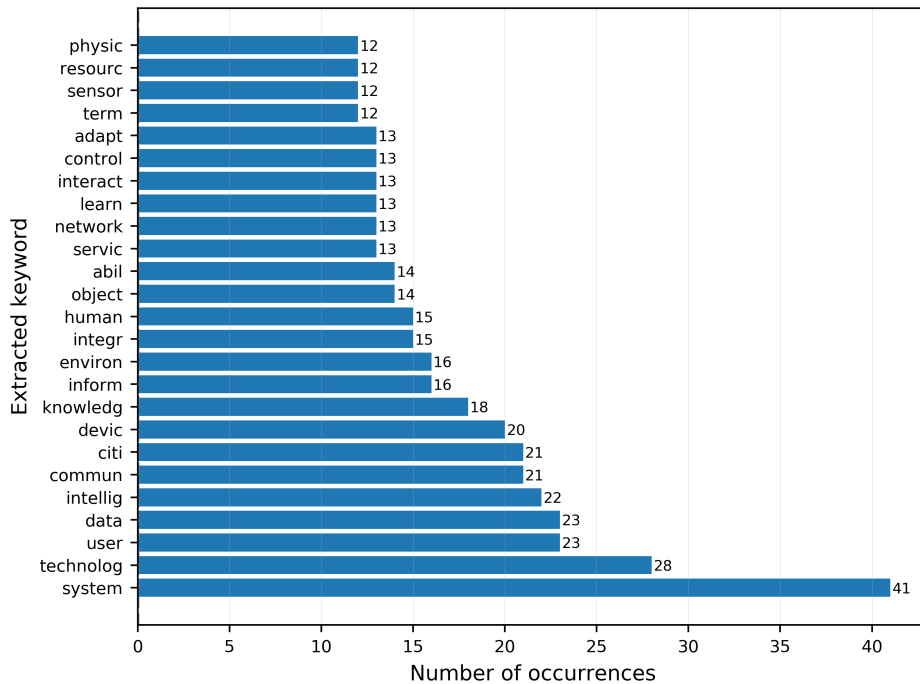


Figure 2.6: Number of occurrences of the top-25 most repeated words considering the full text of the definitions.

Word frequency counting has also been applied to the keywords that were manually extracted from the definitions only. Figure 2.7 shows the occurrences of the top-25 most frequent keywords that were manually extracted from the definitions. Results show that most of the manually extracted keywords presented in this section coincide with the words that have the highest frequency within the full definitions. Aspects such as data, knowledge, environment, communication, sensing, controlling, learn, the presence of an user, and intelligence seem to be relevant within the definitions considering not only in terms of quantity but also considering the qualitative value that we gave to those words at the moment of their extraction. Moreover, there are some concepts that are not present in the results obtained using the full text of the definitions, which have some frequency within the set of extracted keywords such as integration, ICT, development, autonomy, optimisation, infrastructure, and context.

Word clustering

For the Skip-gram model described in Section 2.3.2, the vocabulary size was equal to 100, the embedding vector size was equal to 100, the number of iterations during the training was equal to 50000 with a batch size equal to 100, and a skip window size equal to 5 was used. These hyper-parameters were defined initially considering the literature [199, 200] and empirically fine-tuned afterwards.

Since the word vectors obtained after training the model were high-dimensional, the t-SNE method was applied to aid at performing a visual analysis of the results, which is a technique that allows to visualise high-dimensional data by giving each data point a location in a two or three dimensional map [173]. The learning rate used for t-SNE was equal to 10, with 10000 iterations, and perplexity equal to 3. These values were empirically defined. The K-Means

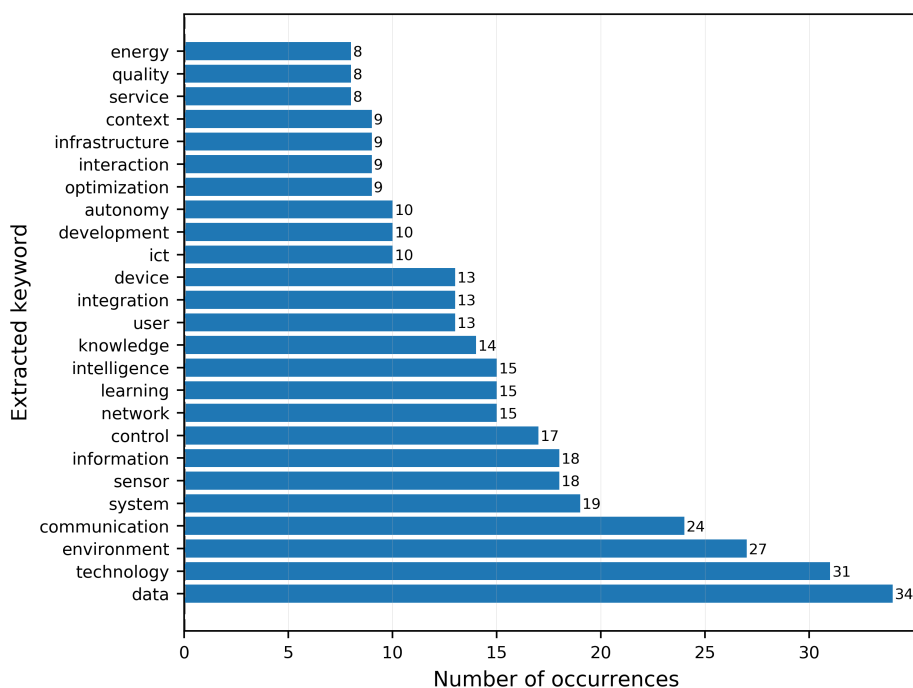


Figure 2.7: Number of occurrences of the top-25 most repeated keywords that were manually extracted from the definitions.

algorithm was also applied to automatically organise the data into representative groups or clusters, using an empirically defined value of $k = 5$, which is the number of clusters to be assigned by the algorithm. K-means presents a number of advantages such as its ease of implementation, simplicity, efficiency, and proven empirical success [178].

Figure 2.8 presents the word embeddings using the first 25 words of the dictionary, which are the most frequent words within the definitions. It is worth mentioning that, same as in Section 2.4.3, the words "smart", "defin" and "use" were removed considering the same reasons. Results show five differentiated clusters. Cluster 1 could be related to the ability of devices to interact and behave in an intelligent manner. Cluster 2 is likely to reflect the use of the word "sensor" with the ability to adapt and to be connected to physical environments. Moreover, humans appear close to these words, which may imply a tendency to use sensors and technological means to sense human actions or data. Cluster 3 does not provide much information since it is the cluster with the less number of words. However, it seems to represent the relationship between services and communication, the later is one of the most frequent words as presented in Sections 2.4.3 and 2.4.5. Cluster 4 has the word "system" in its centre and "object" very close to it, which could seem natural if a system is viewed as a collection of interacting objects or elements. In fact, this interaction (a "network" composed of objects) with the main objective to perform data or information exchange may be reflected by the appearance of the root of both words within the cluster. On the other hand, the root word of "city" is also within the cluster, possibly reflecting a wide use of the term system to describe smart cities. Note that the clusters 3 and 4 could be merged, since they appear relatively close and both of them tend to represent communication capabilities to exchange data or information within a system. Finally, cluster 5 is specifically related to some capabilities of Smart Systems with respect to their environment,

such as "integration", "learning", and "controlling".

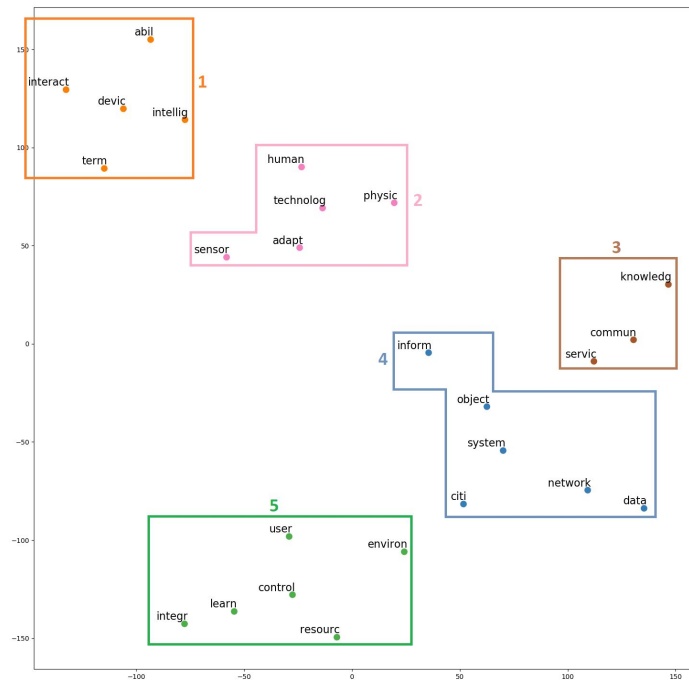


Figure 2.8: Visualisation of the embedding vectors using the 25 most words in the definitions.

Note that the results provided by the Skip-gram model are the output of a fully unsupervised learning approach. Thus, no human direction towards a specific or correct result was provided to the model during training. Therefore, they are often open to different levels of interpretation. However, given the obtained results, the patterns discussed in this section could reflect some specific elements that are described within most definitions of Smartness or Smart Systems.

2.4.4 Smartness and Intelligence

The definitions of *intelligent* and *smart* found in the papers are diverse. Some works treat them as synonym, for instance, [201] considers that a smart system is also intelligent, stating that "*smart is intelligence – the ability to learn or understand or to deal with new or trying situations. The ability to apply knowledge to manipulate one’s environment or to think abstractly as measured by objective criteria (as tests)*". Note that this definition is focused on the behaviour. The work by [202] follows the same tendency, stating that "*an intelligent system is in which different structures are able to co-operate with each other in a coherent way. A smart system is an intelligent system in which services can be exploited by users to their maximum*". However, in this case, the definition is more related to the structure than the behaviour.

Other works consider them as different or complementary concepts. In the work by [142], for instance, it is stated that "*Smart Systems meet to some extent unexpectedly the expectations of their users when they provide comfort or do some daily magic; they sense and act*", whilst "*intelligent systems solve problems in a rational way like humans do and are able to reflect and explain their threads of inference; they solve and justify*". The definition refers to the behaviour of the systems, stating that smartness includes only the capability to sense and act, whilst intelligence allows to reason and explain the why the system performs actions. [203] presents intelligence as an evolution of smartness, stating that "*the future of Intelligent Manufacturing will*

not stop at the level of smart, but to truly realise the intelligent. Intelligent Manufacturing system achieves autonomous learning, autonomous decision-making and continuous optimisation".

Finally, some papers refer to intelligence as a component that must be embedded into systems in order to make them smart. For instance, [204] consider that "sensors, machines, equipment, products, etc., are equipped with embedded local intelligence – which makes them smart objects –and are invisibly (over the cloud) interwoven in order to cooperate and negotiate with each other, thus becoming capable to reconfigure automatically themselves (through actuators) for flexible production of multiple types of products". Moreover, [205] describe smart objects within the IoT from the same point of view: "objects in IoT are made smart through embedding intelligence using some innovational technologies". However, the authors could be possibly describing the concept of *knowledge* when they refer to *intelligence* in their definitions.

2.4.5 Similarities within definitions

The final analysis of the definitions was focused on describing the most frequent words for each domain and their relationships. With this approach, the objective was to extract aspects from Smart Systems that could be considered as common to all domains. Table 2.4 presents the words that were present in the top-25 most frequent words (within the extracted definitions) for at least three different domains. It is worth mentioning that the words are counted after the pre-processing phase described in Section 2.3.2.

Table 2.4: Words, from the top-25 most frequent ones, that appear in the definitions from at least 3 domains. BM: Business and Management. IT: Information Technology.

Word	Domain										Total
	BM	Chemistry	Education	Electronics	Energy	IT	Infrastructure	Manufacturing	Theory		
technolog	X		X		X	X	X	X	X		7
environ	X	X	X			X	X	X			6
system				X	X	X	X	X	X		6
inform	X		X		X			X	X		5
commun			X	X	X	X	X				5
intellig				X		X	X	X	X		5
data	X		X		X	X					4
network	X				X	X	X				4
sensor	X		X	X		X					4
adapt			X	X	X				X		4
user						X	X	X	X		4
servic	X					X			X		3
object	X						X		X		3
self	X			X		X					3
devic	X				X	X					3
knowledg			X			X			X		3
interact			X			X		X			3
control				X	X				X		3
technic				X	X			X			3

Results show that 21 root-words appear within the most frequent root-words of the full definitions of at least three domains. Words related to technology (root "technolog") are the most frequent, appearing within the top-25 most frequent words in seven out of nine domains. This could reflect the relevance of the use or application of technology to achieve smartness. Words related to environment and system are in the second place, appearing in six domains, possibly implying the relevance of a system and its behaviour within a certain environment.

Communication, information, data and network are also other frequent words that are related to the interaction between elements, which is a concept that is commonly addressed in works describing Smartness or Smart Systems.

Besides the raw calculation of root-word appearances, a weighting mechanism has also been applied, aiming at giving more relevance to frequent words that appear in domains with more definitions. The reasoning behind this is that a word that is the most frequent within a very large text corpus will likely be more relevant and descriptive than one that is the most frequent in a much smaller text corpus. Naturally, very simple words are excluded as well as stop-words to apply this reasoning. Hence, weights are given to each domain reflecting the number of unique words within the full set of definitions for that domain.

Therefore, the relevance θ_i of the word i is defined by $\sum_{j=1}^N \lambda_j(i)$ for a set of N domains. The function λ_j is defined considering the number of unique words within the definitions for the domain j , and it is proportional to the domain with the highest number of unique words. Moreover, it considers if the word i is present within its top-25 most frequent words for the domain j , if it is not present, the value of λ_j is equal to zero. The formulation is presented next:

$$\lambda_j(i) = \begin{cases} u_j/u_{max}, & \text{if word } i \text{ is in the top-25 of domain } j \\ 0, & \text{otherwise} \end{cases}$$

Where u_j is the number of unique words within the definitions of the domain j and u_{max} is the number of unique words of the domain with the maximum number of unique words. For instance, the domain *Infrastructure* ($j = 7$) has the highest number of unique words within its definitions ($u_7 = 478$) so $\lambda_7 = 1$, whilst $\lambda_3 = 0.36$ for the domain *Education*, which has a total of 173 unique words. Figure 2.9 shows the results of this analysis.

The results of the weighted frequency calculation are similar to the raw results presented in Table 2.4 since "technolog" is in the first place and "technic" is still the least relevant word. However, the order of some words is different, which may be relevant to select specific attributes to formalise the characteristics of Smart Systems. For instance, with the weighting method, the word "user" is more relevant than only considering its raw appearance in top-25 for each domain. This could reflect the importance of the interaction between Smart Systems and the users. Moreover, some definitions are already based on systems that are user-centred [206–208].

2.5 Characteristics of Smart Systems

Considering the results obtained in this work, a set of characteristics of Smart Systems were extracted. The wide use of the word "system" within the definitions that were found in the literature shows a tendency to consider a systemic view to provide definitions. This systemic view allows to take into account some pre-defined properties or elements of Smart Systems derived from the General Systems Theory (GST) [78], characterised by the presence of system elements, relations between elements, the objective of the system, an external environment and the interface. However, specific attributes that make systems smart do exist, making Smart Systems a specific type of system. These characteristics are presented next:

- **Communication Capability:** the elements must be able to interact to exchange data, communicate their capabilities, and inform about the state of their surroundings [209–215]. Communication is essential for interoperability between the system elements. Note that the words "data", "information", "interaction", and "communication" are widely used in the definitions, reflecting the value of the interoperability between the elements of Smart Systems, or between the system elements and their environment.

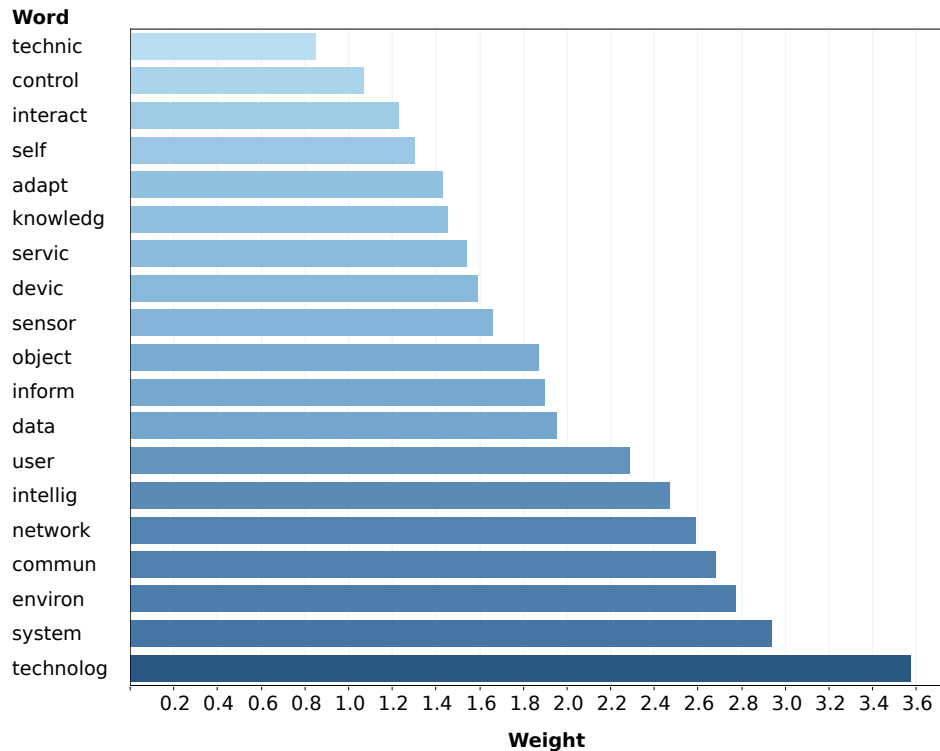


Figure 2.9: Weights of the most frequent words in the definitions considering their appearance in definitions of each domain.

- Embedded Knowledge:** human experience and expertise must be captured by the system [216] as well as information about the environment. In the context of Smart Systems, knowledge can be viewed in different ways depending on the needs of the system. For instance, it can be a knowledge base devised by human experts, common in Knowledge-based Systems (KBS) [217], used by an inference engine to make decisions (model-driven resource). On the other hand, it could be a Machine Learning (ML) [218] model used by an element when it is necessary to process some input data or to retrain the model with novel training samples (data-driven resource). Independently of how it is represented within the system, all of the decisions of the system must be based on knowledge [219].
- Learning Capability:** learning comprises the application of diverse methods and algorithms that allow to modify the knowledge of the system [220–222]. This capability enables an adaptive behaviour, allowing to satisfactorily handle new contexts or situations [223]. Learning is accomplished by the system in an autonomous manner, updating its knowledge with minimal or any external intervention.
- Reasoning Capability:** Smart Systems must be able to reason using their knowledge to make decisions [223, 224]. A range of techniques, such as neural networks, fuzzy logic, evolutionary algorithms, expert systems, etc., can be used by the system to provide strategic decision-making and flexible data processing [225]. Reasoning also allows the system to predict future states of the environment in order to act before the changes are manifested [226].

- **Perception Capability:** The systemic view described earlier implies the presence of an environment that Smart Systems interact with. To be able to perform this interaction, sensors (perception) are used to perceive the state of the environment, and actuators (control) are used to take actions to modify that state. According to the quantitative results obtained in this work, both "sensor" and "control" are words with high level of relevance for describing Smart Systems. Smart Systems must continuously sense or perceive the environment and themselves [139, 142, 214]. This would not be possible without sensors able to monitor and detect changes in their surroundings [205]. Smart Systems are able to describe and analyse the environment using data acquired by the sensors [227].
- **Control Capability:** Smart Systems are ultimately devised to serve humans or to provide the necessary tools and mechanisms to improve aspects of human life. Although this may seem implicit for any human creation, the definitions in the literature explicitly address this aspect. Moreover, they introduce two paths regarding the relationship between users and Smart Systems. The first one refers to the Perception Capability, due to that they are often devised to collect data from users through the use of sensors. The second one is based on a high automation level that allows to avoid the need of user intervention to perform certain tasks or to make some decisions. The control capability is available through actuators that execute decisions made by the system, using the available resources, on the environment in order to modify its state [201]. Smart Systems must have actuators that perform independently without the intervention of elements that are not within its boundaries.
- **Self-Organisation:** the actions performed by Smart Systems are not limited to the environment, since they can also be used to act on the system elements in case that self-organisation is necessary [204]. Hence, the system is able to modify and manage its structure while keeping its original objective [228]. In this context, self-organisation implies the capability to independently adapt since there is no need of any external influence to organise the system elements.
- **Context-Awareness:** the context comprises information that characterises the current state of an environment [38, 229, 230]. A system is self-aware if it considers the context in which it operates when performing its activities [38]. In order to be context-aware, a system requires to sense the environment and to have embedded knowledge that allows it to interpret the state of that environment.

Note that the interaction between the elements of a Smart System produces the emergence of new characteristics or properties. Emergence is based on the idea that a system has one or more properties that are not possessed by its elements individually [231]. Hence, novel characteristics are derived from the interaction between the system elements, including the environment [232]. Emergence derives into unexpected properties that can be positive or negative [233]. For Smart Systems, unexpected behaviours produced by emergent properties must be always positive [142]. In this work, Self-Organisation and Context-Awareness are considered as emergent characteristics.

This set of characteristics provides insight regarding the nature of Smart Systems, which can be considered as systems that are able to update their internal knowledge used for reasoning to make optimal decisions. For this purpose, Smart Systems must have the capability to continuously perceive the environment and to control it, and to allow communication between their elements.

On the other hand, concerning the concept of intelligence in Smart Systems, there are several views of the relationship between smartness and intelligence, as discussed in Section 2.4.4. Intelligence represents the combination of the following smart capabilities: Learning, Reasoning, and Embedded Knowledge. Moreover, intelligence comprises the ability to create or update knowledge (learning) considering the data exchanged between the system elements and the different states that the environment assumes. It also allows the system to adapt to new objectives and states of the environment through the reasoning capability, which is possible using the knowledge.

A graphical representation of a Smart System taking into account the characteristics described earlier is presented. Figure 2.10 shows a meta-model of a system extended with elements able to provide smart capabilities. The model was devised using a Unified Modelling Language (UML) Class Diagram [234]. The language was extended through dotted lines and boxes representing properties that emerge from the relationships between some elements.

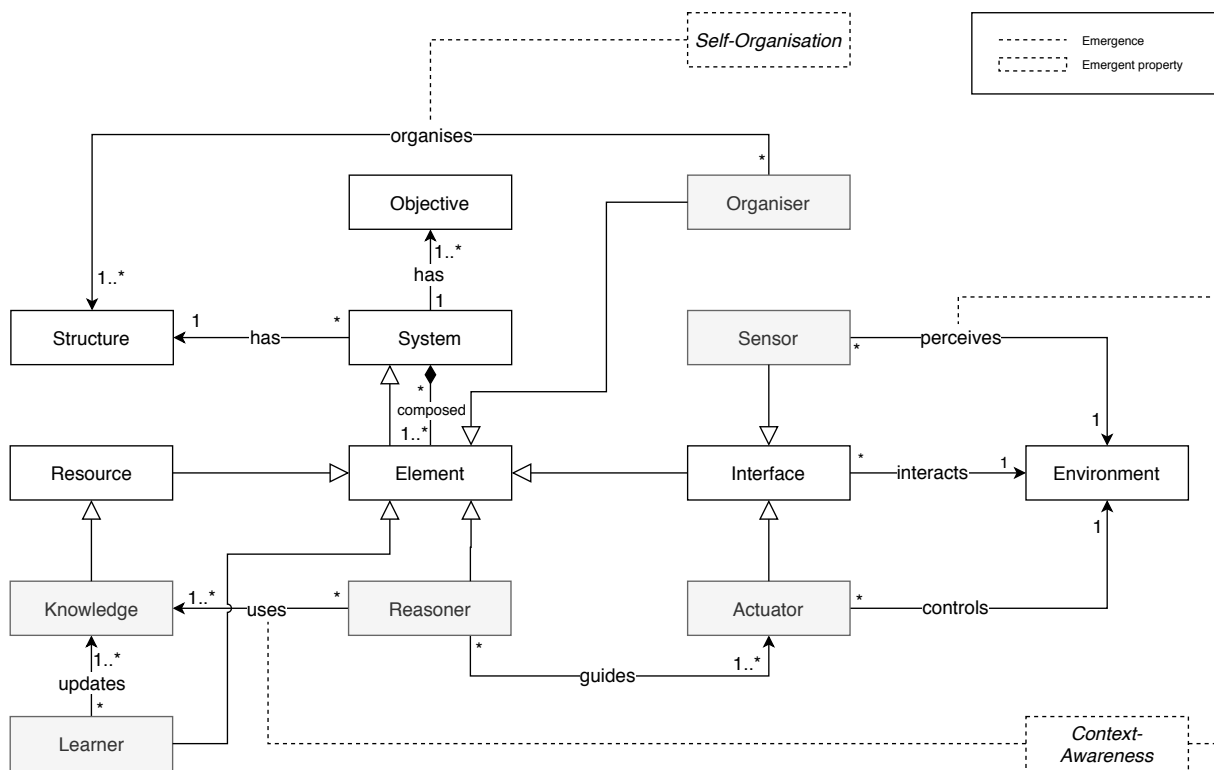


Figure 2.10: Meta-model of Smart Systems.

The model includes basic elements from the GST introduced in Section 2.2.1: the System has an Objective, a Structure, and it is composed of a set of Elements. These Elements could be Resources or Interfaces that are connected with the Environment. Note that the Structure of the System is also considered, which defines the form of the system. It is a schema of the organisation of the elements of the system and their relationships. To provide smart capabilities, the model is extended through the Organiser, which is a system element able to organise the Structure of the System. The Knowledge element is introduced among the resources, which can be updated by a Learner element and it is used by the Reasener. These elements provide the Learning and Reasoning capabilities, respectively. On the other hand, there are two types of Interfaces in Smart Systems: the Sensor and the Actuator. The first one provides the Perception capability whilst the second one allows the system to be able to control the environment (Control

capability). Note that there are emerging properties that are present within this setup as well. Self-organisation, through the interaction between the Organiser and the Structure; and Context-awareness, through the interaction between the Sensor, the Environment, the Reasoner and the Knowledge elements.

2.6 Threats to the Validity of the Literature Review

In the scope of the development of this work, some threats to validity were identified. One of these threads is the fact that the papers obtained by applying the SLR are directly dependent on the search strings used to perform the search in the databases. Hence, a number of relevant papers containing explicit definitions of some Smart System may not have been obtained due to the structure of the original search strings. Another threat to validity is the fact that domain-specific characteristics that could actually provide relevant information for systems within a specific context were discarded. However, the idea was to obtain a generic definition of the characteristics of Smart Systems based on the definitions presented in the scientific literature, thus this could not be considered as a gap, since it is not within the scope of the work. Notwithstanding, future research may address the inclusion of more domain-specific definitions in order to instantiate the generic characteristics defined in this work for more specific purposes. Indeed, new search strings should be defined to perform the literature searches. Finally, the manual selection of keywords within the definitions could represent another drawback. Although a qualitative analysis to obtain relevant keywords from each definition is the core of the methodology, and despite the fact that some relevant characteristics were gathered through this technique, having a human analysing several documents containing a large amount of text could lead to subjective errors. However, the use of the quantitative analysis techniques allowed to attenuate this issue.

2.7 Smart Capabilities in the Context of Design Science

Both the list of characteristics and the meta-model of smart systems presented in Section 2.5 are considered as research artefacts obtained as result of the application of the DSR methodology. Within the **Relevance Cycle**, the main requirement was the definition of the elements that compose a Smart System and their relationships, in order to serve as base for the development of the SAF. The **Rigour Cycle** considered the papers that were gathered through the SLR introduced in Section 2.3. Moreover, the information related to GST, the literature related to the term "smart" in diverse domains, and the related work presented in Section 2.2 were also considered as relevant elements that were treated as part of the knowledge base for DSR. Finally, the construction of the artefacts within the scope of the **Design Cycle** was based on the SLR and the analysis of the gathered papers using the quantitative approaches described in Section 2.3. Modelling concepts and methodologies were also applied for the construction of the meta-model, which was qualitatively evaluated to check its correctness in terms of semantic and syntactic. Note that the ultimate validation metric for the artefacts is based on assessing if they effectively serve as part of the knowledge-base used for the development of the SAF or any other Smart System. In this sense, it can be considered that there exists a validation pipeline: if the SAF is validated, then the knowledge-base used to build it is validated, thus the artefacts introduced in this work can be also considered as valid.

2.8 Conclusion

This chapter described a Systematic Literature Review that aimed at gathering papers containing definitions of Smartness and Smart Systems considering different domains. The objective of the work was to draw common aspects that could serve as basis for a characterisation of Smart Systems from a generic perspective. Text mining methods were applied to obtain quantitative insights from the definitions that were extracted from the papers. Finally, the characteristics of Smart Systems were presented and described. A qualitative analysis of relevant elements that were identified during the review of the collected papers was also presented.

Results showed that the literature often rely on a systemic view to define Smart Systems. Moreover, aspects such as technology and communication are bounded to the description of this type of system including other capabilities such as knowledge, learning, adaptation, sensing and actuation. These aspects were gathered and organised so as to develop a characterisation of Smart Systems that could be considered independent from the application domain. Considering the results, Smart Systems, besides some base characteristics that are common to any system, are able to self-organise, be aware of the context, and support communication between the system elements. Moreover, they are able to learn, reason, perceive themselves and their surroundings, control their environment, and have embedded knowledge to be used for making decisions.

Future work could aim at exploring a higher quantity of papers addressing Smart Systems from different domains. Moreover, concepts regarding Smart Systems should be formally represented, for instance, using ontologies. Implementations of prototypes of Smart Systems for different purposes based on the characteristics presented in this chapter could also be performed. On the other hand, the characteristics described in this work were obtained in a fully data-driven manner and they were devised to be purely atomic; hence, further qualitative analysis of the collected papers could be performed in order to isolate more complex or emergent characteristics.

Finally, from the perspective of the main objective of this thesis, the set of smart capabilities (including their relationships) that were identified and modelled in the work presented in this chapter, acts as one of the cornerstones for the development of the SAF, which is introduced in the following chapter. Indeed, the set of smart capabilities is considered as one of the main sources of knowledge for the development of the framework within the Rigour Cycle of the DSR methodology. Likewise, both the conceptualisations and the smart assessment meta-model act as meta-artefacts that were derived following the same methodology, as explained in Section 2.7.

The Smart Assessment Framework

Contents

3.1	Introduction	39
3.2	The Smart Assessment Meta-model	40
3.3	Meta-model Specialisation: Interoperability Assessment	42
3.4	The Smart Assessment Framework (SAF)	44
3.4.1	Data Perception Service	45
3.4.2	Organisation Service	47
3.4.3	Presentation Service	49
3.4.4	Data Flow in SAF	50
3.5	SAF Case Study: Business Process Capability Assessment	51
3.6	The Conceptual Models in the Context of Design Science	53
3.7	Conclusion	53

3.1 Introduction

This chapter introduces a formalisation of the Smart Assessment process through a meta-model describing its elements and their relationships [87]. The meta-model serves as ground for the Smart Assessment Framework (SAF) [235], a conceptual framework devised to serve as basis for the development and implementation of enhanced assessment systems with smart capabilities such as reasoning, learning, and data perception. These capabilities were defined and conceptualised in the previous chapter. The core concepts related to enterprise assessments are also considered for the development of both the meta-model and the SAF, with special focus on the content of well-known international standards addressing enterprise assessment. The work presented in this chapter is focused on answering the main Research Question (RQ) of the thesis:

- What are the elements that compose a smart enterprise assessment framework?

To answer the RQ, the Design Science Research (DSR) method is followed with the objective to design and evaluate both the Smart Assessment Meta-model and the SAF following an iterative cycle. The concepts and relationships defined in the meta-model are represented using the Unified Modelling Language (UML) [234]. Specifically, the modelling elements from the Class Diagrams of the language are used. A meta-model is employed to represent knowledge since meta-models allow to graphically describe general concepts and their relationships providing a clear view of

those concepts [236]. On the other hand, the SAF is represented using the ArchiMate modelling language [237], which provides means to describe, analyse and visualise the architecture of an enterprise. This chapter presents two illustrative case studies. The first one is a specialisation of the Smart Assessment meta-model focused on Enterprise Interoperability assessment. The second one has as objective the instantiation of the SAF to perform Process Capability assessment.

This chapter is organised as follows. Section 3.2 presents the meta-model for Smart Assessment. Section 3.3 describes a specialisation of the meta-model for Enterprise Interoperability Assessment. Section 3.4 presents the SAF. Section 3.5 introduces an illustrative case study for an implementation of SAF to perform Process Capability Assessment. Section 3.6 describes how the meta-model and the SAF fit in the context of the DSR method. Finally, Section 3.7 enumerates the conclusions and future research perspectives.

3.2 The Smart Assessment Meta-model

Smartness as a concept is associated to some characteristics that could enable improvements of the functioning of certain systems. It is argued that providing smart characteristics to the assessment process could enhance the diverse activities and sub-activities that are performed during the overall process. The meta-model for Smart Assessment is shown in Figure 3.1. The different elements of the meta-model are detailed next:

- The core of the meta-model is the Assessment Process, which assesses an Assessed System considering an Assessment Scope defining the boundaries of the assessment as established by the stakeholders.
- The Assessment Process uses an Enabling System to properly carry out the assessment activities. An Enabling System is responsible to support a System-of-Interest [238], which is the system of interest of an observer [239]. In the meta-model, the System-of-Interest is the Assessed System. Both systems, Assessed and Enabling, are sub-types of the System element, which is able to use and produce the Resource element.
- Two types of Resources are considered: Data and Knowledge. The Data element defines some Attribute to be assessed and it is the input of the Enabling Systems to ultimately define the Assessment Result. In addition, the Knowledge element serves as basis for decision-making regarding the final Assessment Result.
- The Assessment Result is composed of one or more identified Problems and the proper Solution to solve those problems.
- The Enabling System could be of different types such as Learning System, Reasoning System, and Measuring System. The Learning System is capable of producing or updating Knowledge, whilst the others are focused on using it to provide results.
- An Organising System is also included with the purpose to manage the Measuring, Learning and Reasoning systems.
- The Assessment Process uses the Assessment Framework element, which is an element composed of a set of Assessment Indicators and one or more Measurement Mechanisms. The first one is a reflect of the To-Be of the Assessed System whilst the second one defines means to measure the Assessment Result. Moreover, the Assessment Process analyses some Characteristic of the Assessed System, which subsumes its Attributes.

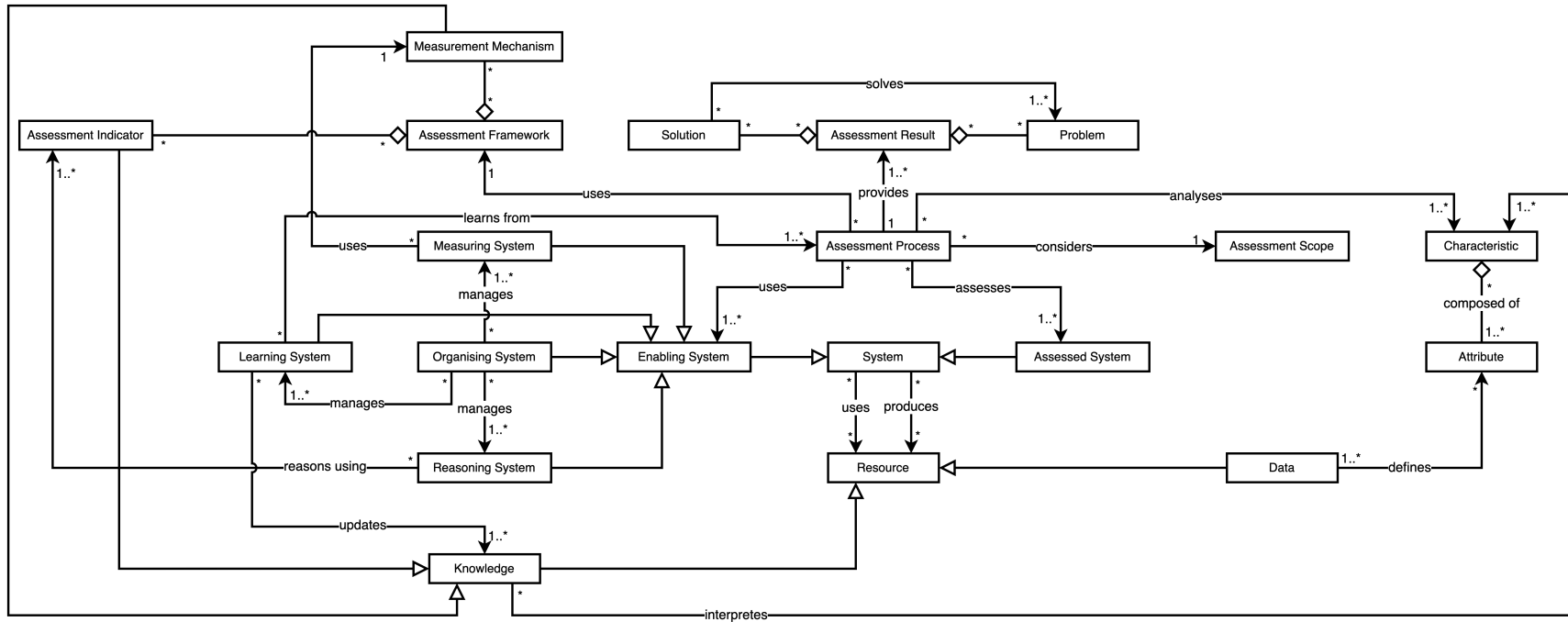


Figure 3.1: Meta-model defining the components of a Smart Assessment.

3.3 Meta-model Specialisation: Interoperability Assessment

Organisations have the necessity to interoperate in order to share information and achieve common and separated objectives [77]. Interoperability refers to the ability of enterprises to interact, and research in the field is mostly based on removing interoperability barriers [240]. Interoperability can occur between the following organisational layers: data, services, processes and businesses [240]. Considering the process layer, interoperability pursues to make various organisational processes collaboratively work in a standardised manner [240]. Moreover, the literature presents three ways to relate systems in order to interoperate: integrated approach (a common format is defined for models), unified approach (a common format exists but at the meta-level only), and federated approach (there is no common format for models) [241].

A specialisation of the meta-model described in Section 3.2 is introduced with the objective to explore the capability of the meta-model to adapt to specific needs. The specialised model is focused on Enterprise Interoperability. The objective is to define the Maturity Level of an Enterprise regarding Interoperability.

The new elements introduced for the specialised meta-model are defined specifically in the Ontology of Enterprise Interoperability (OoEI) [242], the Maturity Model for Enterprise Interoperability (MMEI) [24], and the Ontology for Interoperability Assessment [10]. Brief descriptions of each novel component of the specialised model considering the Interoperability scenario are presented next:

- The specialisation includes an Enterprise as a type of Assessed System.
- The Enterprise is evaluated considering the Evaluation Criterion element, which is a type of Assessment Indicator for Interoperability.
- The Characteristic to be evaluated is Interoperability, which is qualified through a Maturity Level provided as a part of the Assessment Result.
- The Interoperability Area is considered as an Attribute qualified through an Area Maturity Level.
- The Maturity Level reflects the global Interoperability Maturity of the Enterprise and it concerns a set of Area Maturity Levels qualifying some Interoperability Area.
- The element in charge of calculating the maturity levels (for Interoperability Area and Enterprise) is the Measurement Mechanism, which is a schema characterising the methods to provide quantitative measure of Interoperability considering the Maturity Model that is used for the assessment.
- In addition to the Maturity Level and Area Maturity Level elements, the specialisation includes the Best Practice and Interoperability Barrier elements as part of the Assessment Result. The former is a type of Solution to solve the latter, which is a type of Problem.
- Each Evaluation Criterion references one or more Best Practices that allow to remove an Interoperability Barrier. Hence, there is a direct link between an Evaluation Criterion and Interoperability Barriers that are solved through the application of Best Practices.

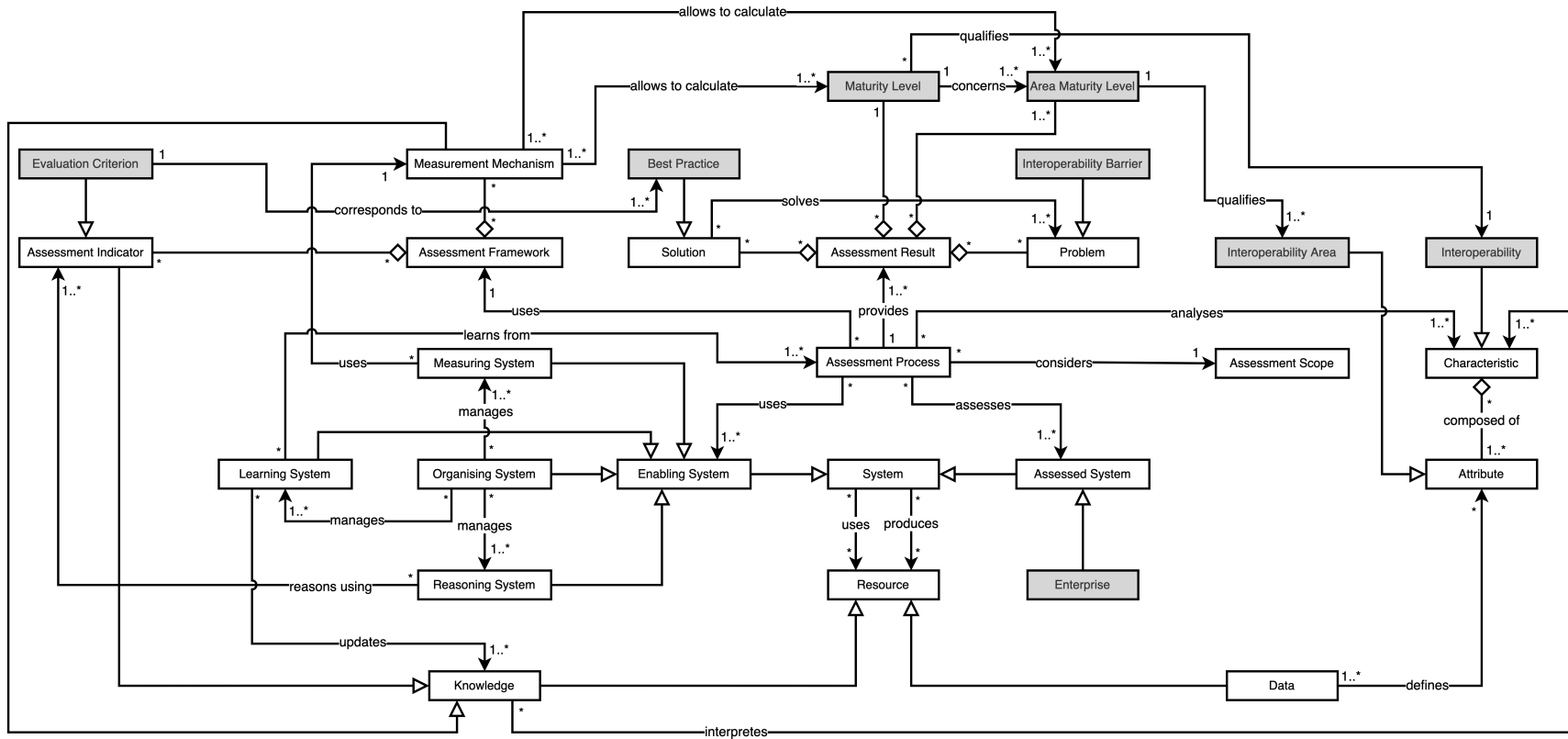


Figure 3.2: Smart Assessment meta-model specialised for Enterprise Interoperability.

3.4 The Smart Assessment Framework (SAF)

The Smart Assessment meta-model, introduced in Section 3.2, describes a collection of elements from diverse categories. For instance, the Reasoning and Learning systems are strongly linked to technological implementations, whilst the Assessment Process and Assessment Result elements of the meta-model are more related to the business side. Although this provides a complete overview of how a smart assessment could be performed considering all the parts that are required to be put in place to allow the emergence of smart capabilities when performing assessments, it does not allow to clearly separate concerns. In this sense, the SAF could be considered as an evolution of the meta-model, with a strong focus on two aspects: distributing the elements of the conceptual model into different layers, and reducing the amount of elements so as to have a simpler and more concrete view of how a smart assessment system could be implemented. An overview of the proposed SAF is presented in Figure 3.3.

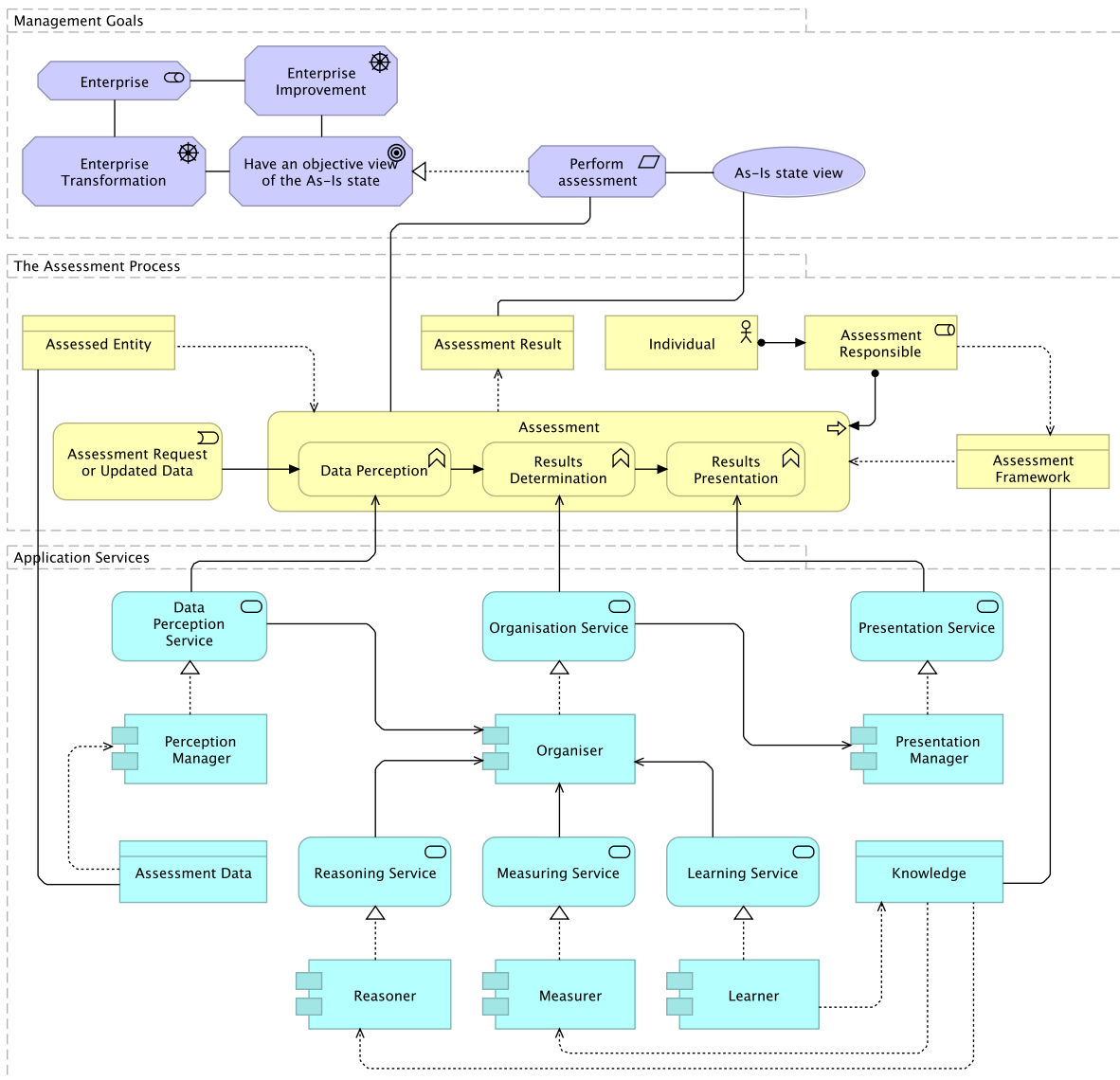


Figure 3.3: Conceptual framework defining the components of a smart assessment.

The framework uses as core the elements of the Smart Assessment meta-model. Furthermore, SAF is strongly focused on the set of smart capabilities defined through the Systematic Literature Review, described in Chapter 2. The ArchiMate modelling language is a standard used in this work to graphically describe the elements of the framework and their relationships. ArchiMate is specifically devised for Enterprise Architecture (EA) to aid enterprise architects to describe, analyse and visualise relationships among architecture domains through simple visual representations of architectural elements [237]. The language is used in this work to represent the elements of the framework since it provides means to distribute architectural elements in layers, providing a heuristic view of the architecture of the framework, reflecting the way that the components of each layer are interrelated. ArchiMate distributes the elements in three core layers: the Business Layer, which defines business services, realised by business processes and performed by business actors; the Application Layer, which depicts application services supporting the business and the application components that perform them; the Technology Layer, which describes the technology services and computer, hardware and software systems to provide those services.

It is worth mentioning that this work does not present an ArchiMate model designed for a specific organisation, instead the modelling elements and the layer organisation provided by the language are used to define and organise the elements of the framework. The proposed model focuses on two of the three core layers of ArchiMate: Business and Application. We do not provide elements of the Technology layer, since it ultimately depends on specific implementations to perform the assessments in enterprises. The Strategy layer of ArchiMate is also employed to represent the drivers, goals and value of the enterprise with respect to the assessment process.

The SAF is composed of three levels: Management Goals (Strategy), The Assessment Process (Business), and Application Services (Application). The first layer of the architecture, from top to bottom in Figure 3.3, shows that an assessment is a requirement to provide a view of the As-Is state of the assessed entity (value). This goal is driven by the need of an Enterprise to transform or improve. The second layer presents the Assessment Process. It analyses an Assessed Entity and it is triggered by a new assessment request or updated assessment data. The process is composed of three main activities: Data Perception, Results Determination, and Results Presentation. It provides an Assessment Result considering data from the Assessed Entity using an Assessment Framework, which provides a reference describing the To-Be state of the assessed entity through a set of Assessment Indicators or Requirements, and the Measurement Mechanism that is used to measure the gap between As-Is and To-Be. Note that the Assessment Process has one or more individuals acting as responsible. The final layer describes the elements in charge of performing the assessment when smart capabilities are used for such purpose. It is composed of three main services devised to execute each activity of the process: Data Perception Service for the Data Perception activity, Organisation Service for the Results Determination activity, and Presentation Service for the Results Presentation activity. This layer is the core of the framework. Hence, the rest of this section is focused on explaining each of these services and their components in detail. Likewise, a visualisation of how the Assessment Data flows and is transformed during its passing through each component of the SAF is also presented in the last part of this section.

3.4.1 Data Perception Service

The Data Perception Service is provided by the Perception Manager component. Its main objective is to sense the data to be used for the assessment, apply some pre-processing steps and providing the prepared data to the Organiser component. Note that, in the scope of the

framework, the Assessed Entity could constantly produce data that could have an heterogeneous nature. This data must be autonomously sensed by the Perception Manager as the Assessment Data component. Note that the means to perform data filtering must be put in place in order to use only the data that is relevant for the assessment.

Perception is the ability of an individual of being aware of the properties that are present in a certain environment through the use of physical senses [51]. In this work, the Perception Manager is considered as an enabler of the perception property that is intended to be exhibited by agents through the use of computational models that are analogous to human perception mechanisms [243]. This perception capability allows to obtain information from the environment that can be further used by an agent to interpret, plan, reason, react, adapt and act on the environment [243].

There are various types of information that can be perceived by an assessment system and used as input to perform an assessment. This work considers a hierarchical perspective of the possible measures that can be sensed, which are proposed in [244] and described as follows:

- At the **Enterprise** level, aspects such as production costs, profit, demand, customer satisfaction, strategies, regulations, business models, among others, can be perceived by the system in charge of the assessment.
- At the **Factory** level, costs, labour hours, worker operation, production capacity, worker satisfaction and the proper production of the factory are possible elements to be sensed.
- In a lower level, the **Shop**, aspects such as equipment utilisation, stock, and processing cycles can be measured.
- At the **Cell** level, tasks progress, equipment requirements, and lead times can be sensed.
- At a lower level, the **Station**, aspects such as machine failures, process time, collision detection can be sensed.
- Finally, at the **Equipment** level, raw measures such as vibration, temperature, displacement, noise, energy, among others, can be obtained.

Note that these initial measures are represented through specific data structures. Raw measures such as temperature, electricity level, pressure, etc., are represented through structured data and obtained directly using physical sensors. On the other hand, customer satisfaction, worker satisfaction, strategies, worker operation, etc., could have various data structures, and in some cases they are unstructured such as text, audio, video, or images. In this context, a smart system capable of performing smart assessments must be prepared to gather and pre-process both types of data (structured and unstructured), in order to produce post-processed data ready to be used by the components of the SAF that provide the Organisation Service.

It is worth mentioning that flexibility is a relevant property of the Perception Manager. In this sense, it could use physical sensors that allow to establish a direct connection to the real-world; software listeners able to consume digitised information or recognise changes in a certain information source that is being controlled; and software tools that allow users to directly input assessment data. Moreover, the Data Perception Service could be provided through the application of a combination of the data gathering strategies mentioned before. If this is the case, the Perception Manager must be able to manage and organise all these data sources in order to prepare them for the Organiser component of the SAF.

The data collected by the elements that are employed by the Perception Manager can be passed directly in their raw format to the Organiser component, or a certain number of pre-processing steps can be applied if necessary. Taking this into account, three possible levels of data treatment at the level of the Data Perception Service are defined, this separation is based on the level of intelligence employed by the manager to treat the data:

1. **Raw data gathering.** Focused only on gathering the required data and hand it over to the Organiser component with no previous processing.
2. **Edge data pre-processing.** Some pre-processing functions can be applied on the gathered data by the components used for perception before being passed to the Organiser. Some of the operations include elimination of duplicates and outliers, filtering functions so as to keep only the required data, and formatting of text, dates, features, etc. This concept is strongly based on the edge-computing paradigm [245], which has as main idea the use of the processing and storage capabilities of the distributed nodes to improve response times and bandwidth.
3. **Edge intelligent computing.** A certain level of intelligence is already present in the components that gather the data, which is processed by those components to obtain partial results that are then passed to the Organiser component to apply other algorithms or to perform some aggregation operation. Depending on the problem to be addressed, these edge components could have a high level of complexity. In this sense, they could even be considered as sub-smart assessment systems with most of the components of the Application layer of SAF, thus resulting in a system of systems [238].

Each level is built on top of the lower one. Thus, in order to perform Edge intelligent computing, the data must have been first gathered and pre-processed; and the same scenario applies for Edge data pre-processing. Therefore, each level represents how much the data is treated before being passed to the Organiser component and how much the original data is modified: from not being treated at all, to being a result of the application of one or more data-processing mechanisms.

3.4.2 Organisation Service

The Organisation Service is the core of the Application layer. It is performed by the Organiser component and its sub-components. The Organiser receives the assessment data (pre-processed or not) from the Data Perception Service and it distributes it to three components depending on the specific needs: Reasoner, Measurer, and Learner. The Reasoner fulfils the Reasoning Service, and it uses Knowledge to infer new information regarding the Assessed Entity. Note that part of the Knowledge is an Assessment Framework, which defines qualitative and/or quantitative aspects to provide the results of the assessment. The Reasoner is devised for automatically inferring Knowledge through reasoning mechanisms [246]. For this purpose, it analyses a set of Assessment Indicators within the Assessment Framework along with the Assessment Data to infer new knowledge to be provided as part of the Assessment Result. The Measurer considers the Measurement Mechanism defined by the Assessment Framework to calculate and provide quantifiable results. Moreover, if score aggregation is necessary and if some aggregation method is defined by the Assessment Framework, the Measurer must be able to perform this task. This means that the Assessment Framework can not be only for the Reasoner but it is available to all the components of the Organiser. Finally, the Learner provides the Learning Service with the purpose of updating the Knowledge. The tasks performed by the Learner are typically performed manually. In this scenario, one or more human experts learn from the assessment

evidence, the entire process that was followed to perform the assessment, the results that were obtained. The lessons learned from the experience of performing the assessment are then used to modify the current Knowledge representation (which includes the Assessment Framework) so as to perform better assessments in the future. This is an iterative cycle in which the Knowledge is refined with each iteration. However, it is argued that automatic learning could also be performed by dedicated systems. Future research should target the design, development and evaluation of different types of autonomous Learner components that could be able to update the Knowledge so as to perform future assessments with more efficiency, robustness, quality, etc. As an intermediate step towards a fully automatic learning mechanism, the initial representation of the Knowledge could be defined by human experts and future updates and modifications of its content could be performed by a dedicated system. It is also worth mentioning that the level of complexity of updating existing knowledge depends on the means used for representing that knowledge. For instance, mechanisms to modify a neural network model that acts as source of knowledge are much different from those used to change elements of an Ontology and inference rules.

An element of high relevance for structuring the Organisation Service is the Knowledge. There are several possible sources of information that can be used or considered as instances of the Knowledge component, and the selection of an alternative depends on the problem to be solved. That is, the scope of the assessment. The Knowledge component could range from simple rules of inference and aggregation functions defined by a software library to pre-trained deep neural networks and hand-crafted combinations of ontologies. The data contained in a database can also be considered as source of knowledge that can be used by an assessment system. Note that the structure and functioning of both the Reasoner and the Learner could change according to how the Knowledge is structured in the system. For instance, from a data-driven AI perspective, a Deep Learning model that was previously trained to perform some task could be an initial instance of the Knowledge component. The Reasoner, in this case, could be a script that loads the model, feeds it with the assessment data and gets its results. Different mechanisms could be used to define a Learner component for this scenario, e.g. incremental learning techniques [247] could be applied to continuously extend the knowledge of the model using new input data. On the other hand, for a model-driven or classical AI perspective, the initial instance of the Knowledge component could be an Ontology describing concepts from an Assessment Framework. The Reasoner could be an inference engine that uses information from individuals of the Ontology to infer the information of other individuals. The Learner, in this case, could be composed of mechanisms that allow to automatically update the concept hierarchy of the Ontology [248]. The Knowledge could also define measurement rules and metrics that can be activated according to the content of the input Assessment Data and could be applied by the Measurer to obtain quantitative results. In this sense, the Measurer component is the most dynamic and prone to change element of the SAF, since it could be considered as a wildcard that can be instantiated to perform all types of calculations that depend on the needs of the problem.

It is worth mentioning that an assessment system that follows the structure proposed in the SAF could use more than one instance of the Learner, Reasoner, Measurer, and Knowledge. These instances could interact with each other and exchange information guided by the Organiser component. For a concrete example of the implementation of the SAF with instances of more than one Knowledge component the reader may refer to Chapter 4. The presence or absence of more than one component collaborating to support the Organisation Service depends on various aspects such as the scope of the assessment, the available Assessment Data, the Assessment Framework that is selected to guide the assessment, etc.

A final consideration regarding the Organisation Service is related to the Assessment Frame-

work, which is part of the Assessment Process layer of Figure 3.3. This element is relevant since the Knowledge component must act as a formal representation of the content of the Assessment Framework that is relevant for the definition of the Assessment Result. Therefore, the proper mechanisms to clearly translate the requirements, indicators, measurement mechanisms, aggregation operations, etc., that are defined by the Assessment Framework into the elements of the Knowledge component must be put in place. Examples of two approaches for translation were already mentioned before in this section: an Ontology could act as a hand-crafted translation of some aspects of the Assessment Framework, as well as some rules for inference, in which case at least the first version of the Knowledge component is explicitly defined by a human expert; whilst a Deep Learning model could be trained to contain implicit information from the Assessment Framework following a data-driven approach.

3.4.3 Presentation Service

The Presentation Service receives the results of the assessment provided by the Organisation Service and generates a structured view of those results to the stakeholders. The Presentation Manager, which is the component in charge of providing the Presentation Service, should provide a view to the stakeholders that is informative enough for decision-making. Moreover, personalisation is a very relevant aspect of this component, since it is necessary to adapt the results to the profile of the individual requiring those results. In this sense, it is relevant to emphasise the gaps between As-Is and To-Be of the Assessed Entity that are relevant to the stakeholders and how the fulfilment of certain quality or performance requirements to close those gaps may have negative or positive impact on other aspects of the entity.

Note that in Section 3.4.1 a perception agent is considered as analogous to the Perception Manager. This is not the case for the Presentation Manager because assessments have a more passive nature: they serve only as enablers for the acquisition of information that reflects the state of an enterprise entity. Therefore, the manager cannot be considered as a form of actuator able to perform actions on the environment since this is beyond the nature of an assessment. Notwithstanding, the results that are defined by the Organiser can be configured in such a way that they can be passed to actuators (or to systems in charge of managing those actuators) through the Presentation Service in order to ultimately allow them to perform actions on the environment. This allows to close the loop Sense-Think-Act, if required, without the necessity to add properties or components to SAF that are beyond its own scope.

There are two types of assessment results that the Perception Manager can provide:

1. **Qualitative.** In most cases, this type of result is composed of text, enterprise models or tables describing the state of the Assessed Entity, which has been analysed from a qualitative perspective using the mechanisms provided by the Organiser component for such purpose. They may provide capability or maturity levels with their descriptions, improvement suggestions, and weaknesses of the entity and its problems. Reports and tables can be used to present this kind of result.
2. **Quantitative.** This type of result is composed of numerical values and their possible visualisations, which are obtained after performing mathematical operations or applying mechanisms supported by the Organiser component and its sub-components. Charts, tables and reports are possible containers of quantitative results. Different techniques for numerical data visualisation can be used to generate the figures such as bar charts, scatter plots, histograms, pie charts, heat maps, radar chart, etc.

The channels that could be used for results presentation are various and choosing one of them ultimately depends on how the stakeholders want to have access to those results. Some examples of presentation channels include the front-end component of web, desktop and mobile applications, direct emails containing reports, tables, graphs, and enterprise models, messages to actuator components from actuation systems, etc. Note that, if the results of the assessment are going to be passed to another system, their format and structure must be defined in such a way that they could be able to be used by that external system. Therefore, specific mechanisms to support interoperability between systems [249] must be put in place as sub-components of the Presentation Manager.

3.4.4 Data Flow in SAF

An assessment performed by a Smart System can be summarised as the transformation of the original Assessment Data into an Assessment Result, such transformation is performed and managed by the different components of the system. Figure 3.4 graphically describes how the Assessment Data flows and is transformed during its passing through each component of the Application layer of the SAF. For this purpose, several examples of possible transformation paths are presented for different types of Assessment Data and technological components of the SAF. Likewise, some possible types of Assessment Results are included in the figure. This allows to illustrate possible application scenarios that can be addressed through different implementations of the SAF, thus showing the functioning of the SAF from a data flow perspective.

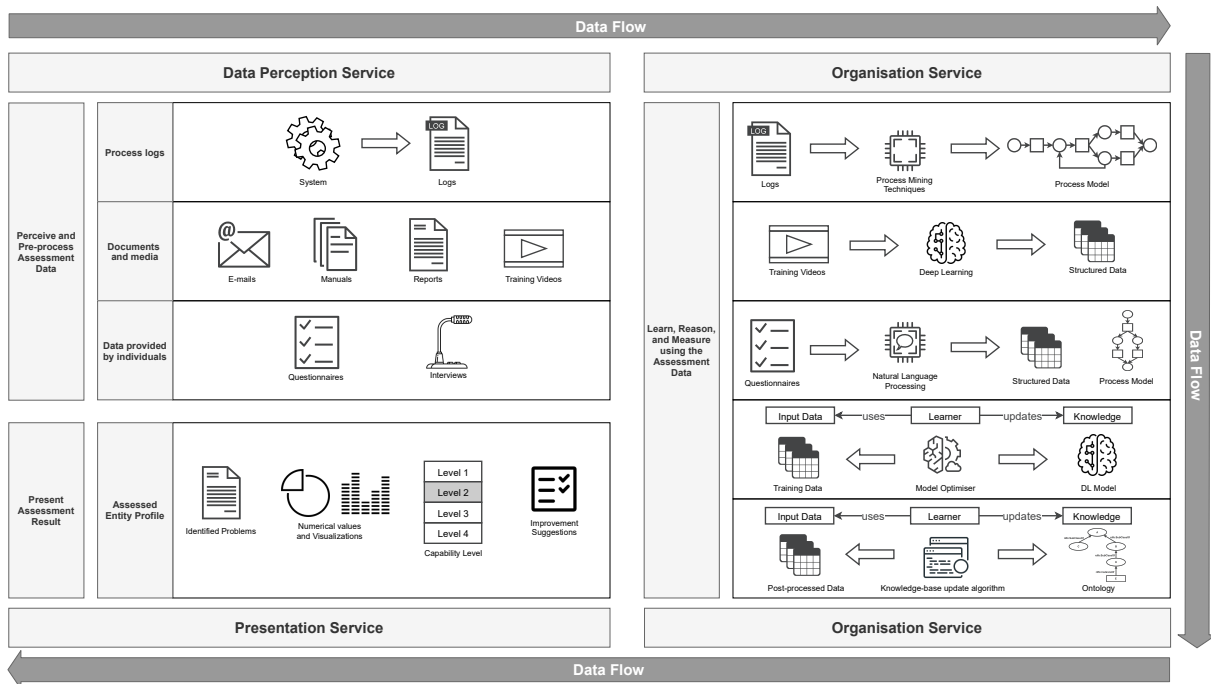


Figure 3.4: Flow of the assessment data through each component of the Application layer of the SAF.

For a given assessment, the data flow commences with the perception of the Assessment Data, which could be of any kind including process logs extracted from a database, documents, textual data, tables from a database, multimedia, etc. This data may be pre-processed by the components of the Perception Manager and then passed to the Organiser component. This element of the SAF

could implement Machine Learning and Deep Learning algorithms, Knowledge-based systems and Inference Engines, process mining techniques, statistical analysis, aggregation operations, etc., to obtain the assessment results in the form of process models, structured data contained in tables, discrete and continuous values, and text containing recommendations and identified problems, among other possible data configurations. Note that the elements that are part of the Knowledge component of the SAF (DL Model and an Ontology in the example) can be updated through various mechanisms such as a model optimiser for Deep Learning models and knowledge-base updating algorithms. The results obtained after the application of these methods are then aggregated, post-processed and passed to the Presentation Manager, which provides the Presentation Service by presenting the results using the appropriate channels for such purpose.

In this context, the Assessment Data follows a transformation flow in which it is converted from raw data into an Assessment Result that can be used as a source of information that could allow to close the gap between the As-Is and the To-Be states of the Assessed Entity. Moreover, as mentioned in the previous section, it could also pass through a full transformation process to achieve the Sense-Think-Act paradigm, resulting in a set of actions that will be performed on a certain environment without requiring the participation of humans in the entire process.

3.5 SAF Case Study: Business Process Capability Assessment

This section presents an example of an implementation of the SAF through an illustrative case study. For this purpose, a Business Process Assessment scenario is considered. Specifically, the process to be assessed is regarding Samples Management (SM) process in an enterprise with the objective to obtain a qualitative view reflected by a capability level and a set of improvement recommendations for the process. First, a brief description of the process is presented. After, the Assessment Framework used to guide the assessment is presented. Finally, the possible instantiation of the SAF is described. Figure 3.5 provides a graphical representation of the elements of the framework used to perform an assessment for this specific case.

The SM process comprises the activities of reception, treatment, destruction, and returning of different types of chemical samples that are used in an organisation. The data describing the process is collected through a series of semi-structured interviews carried out by an assessment team. Since it is common that enterprises request to perform assessments that are aligned with international standards, the interviews are based on asking questions guided by an ISO/IEC 15504-330xx [16, 250] compliant assessment framework named TIPA [66]. The TIPA framework provides two main elements: the Process Assessment Model and the Assessment Method. Both compose the Assessment Framework in the scope of SAF, which includes a set of Assessment Indicators and a Measurement Mechanism defining the method to results calculation. The Assessment Data is composed of SM Evidence gathered by an assessment team that previously carried out interviews to the actors of the SM. This is the only activity of the assessment that is performed by humans, all other services are automatically provided by software components. The SM Evidence, registered in raw files by the assessment team, is composed of partial conclusions from the assessors regarding each Assessment Indicator and it is sensed by the Perception Service each time the files are updated. In the case study, the Perception Service is provided by the SM Evidence Perception from Figure 3.5, which is an instance of the Perception Manager of the SAF. After being sensed, the evidence is pre-processed and passed to the Organiser component.

The Organisation Service performs the assessment. For this purpose, the Knowledge is composed of a set of TIPA-based Assessment Indicators, and a TIPA-based Measurement Mechanism that defines the means to calculate ratings for the process and to match the SM Evidence with

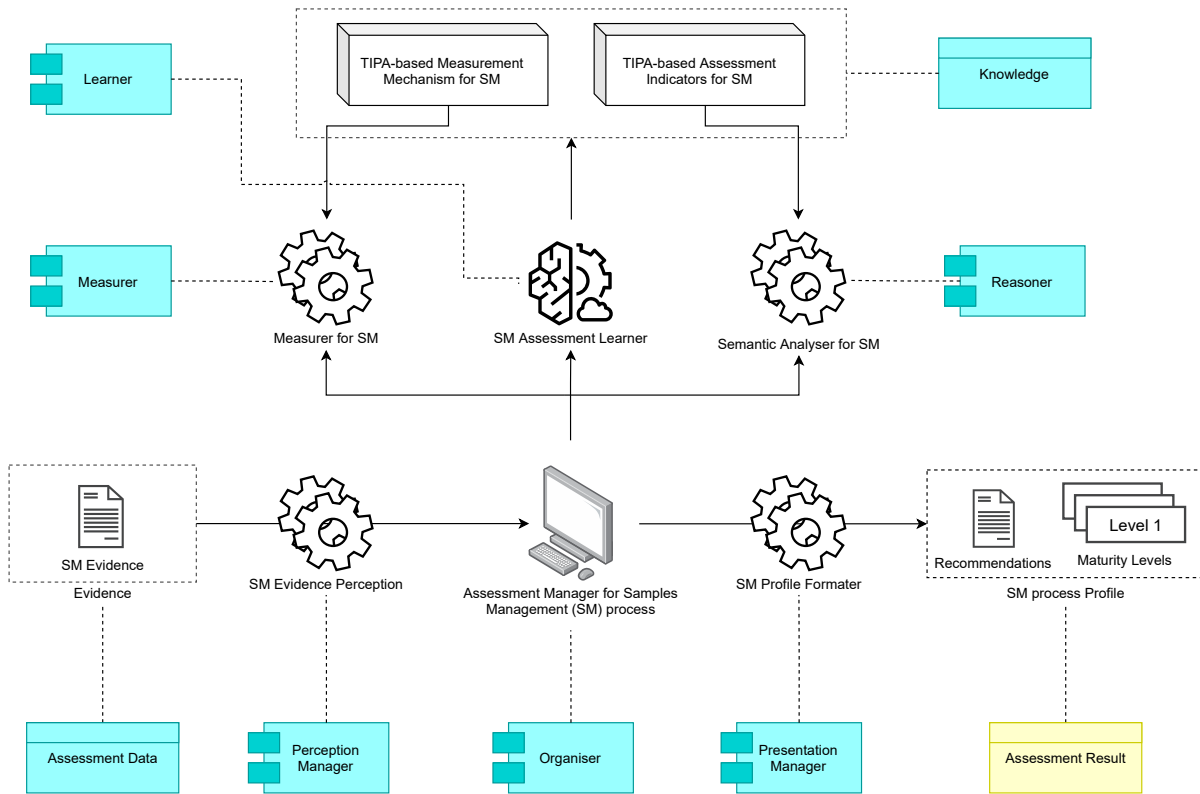


Figure 3.5: Implementation example of the SAF for the assessment of a Smart Samples Management (SM) process using the TIPA framework as Assessment Framework.

the Assessment Indicators. This Knowledge is used by three components: the Reasoning System, which in this case study is a Semantic Analyser able to interrelate each element of the TIPA-based Assessment Indicators with the SM Evidence; the Measuring System, which applies the calculations to produce the rating scores for the SM taking into account the results provided by the Semantic Analyser; and the Learning System, which identifies a set of Assessment Indicators that are not often matched to the SM Evidence considering some threshold, these elements are discarded or updated for future assessments. Note that, in this case, the task of the Learner is simplified so as to be fully automated. However, it could also be composed of human experts able to analyse the overall assessment process in order to update the existing Knowledge with more details.

The SM Profile Formater is an instance of the Presentation Manager. It receives the result provided by the Organiser and formats it to be presented to stakeholders. It provides the rating of the process with capability levels, and a set of final comments in plain text format. Moreover, the interviews carried out by the assessment team could also be recorded and converted into text by the Data Perception Service through the use of Natural Language Processing methods and then be provided to the Organiser to continue the SAF pipeline.

The use of specialised components with smart capabilities could represent an improvement of the assessment process in terms of cost and time. For this case study, analysing the evidence, matching it with the assessment indicators and calculating scores are meant to be performed entirely by the SAF components. Moreover, the gap between As-Is and To-Be could be represented with more fidelity so as to provide improvement recommendations. Likewise, the presence of a Perception Service able to sense changes in the data describing the assessed entity allows to

provide real-time results. Note that technical implementation aspects are dependent on the type of Assessment Data to be perceived by the Perception Manager and the Assessment Framework.

3.6 The Conceptual Models in the Context of Design Science

Both the meta-model and the SAF are considered as artefacts in the context of the DSR methodology. This section presents the descriptions of the elements of each cycle of the three-cycle view of DSR method that were used for the development of the artefacts.

In the **Relevance Cycle**, the main requirement is the definition of the elements of the Smart Assessment meta-model and the SAF and their relationships. For the meta-model, the elements are defined using UML classes [234]. As for the SAF, its elements are grouped considering three layers: Management Goals, Business and Application. Although ArchiMate [237] is used as modelling language for defining the framework, the Technology layer is not considered since it is focused on specific implementations within an enterprise, and this is not within the scope of this work. The **Rigour Cycle** considers the following sources of knowledge for the design of the artefacts presented in this chapter: international standards related to quality, assessments, and software engineering, namely the ISO 9001 [251], ISO/IEC 33001 [16], ISO/IEC 15504 [94], ISO/IEC/IEEE 29148 [252]; the General Systems Theory [253]; the Software Engineering Body of Knowledge [238]; the smart systems meta-model, its concepts, and the smart capabilities presented in Chapter 2; and the scientific literature addressing enterprise assessment. The research activities of the **Design Cycle** include the design and evaluation of the artefacts in an iterative cycle. The design was performed considering the knowledge base from the Rigour Cycle, whilst the evaluations were carried out through meetings with specialists in the field of assessment, specifically within the domain of Business Process Capability assessment and Enterprise Interoperability assessment. Besides the meetings with experts, the evaluation of the artefacts is also based on empirically observing if their structures are robust enough in terms of compliance with concepts from the source of knowledge that are defined within the Rigour Cycle without discrepancies. Finally, the SAF is specifically validated through its implementations, which are presented in detail in the second part of this thesis.

3.7 Conclusion

This chapter presented a meta-model that defines concepts related to the Smart Assessment process and a conceptual framework to guide smart assessments in organisations named Smart Assessment Framework. These two models are considered as artefacts in the context of the Design Science Research methodology. The elements of each artefact are grounded on concepts from the scientific literature, international standards, and the expertise of assessor and researchers. The UML and ArchiMate modelling languages were used to visually represent the elements of the artefacts described in this chapter along with their relationships.

The proposed Smart Assessment meta-model was validated through a case study based on a specialisation of the concepts of the model to support Enterprise Interoperability Maturity assessment. This specialisation allowed to experiment with the generalisation capability of the model. Given the structure of the specialisation, it can be considered that the meta-model provides the proper structure to be specialised for different domains and assessment approaches. Moreover, the model could not only be applicable for organisational assessments but also to a diverse range of contexts. On the other hand, the SAF was validated through an illustrative case study describing a Business Process Capability assessment scenario. Each element of the

framework was linked to specific entities to describe how an assessment could be performed within the scope of the SAF. This case study showed that it is possible to derive specific assessment pipelines supported by components with capabilities such as reasoning, learning, and embedded knowledge. The content of both, the meta-model and the SAF, allowed to answer the Research Question introduced in this chapter, which referred to the conceptual elements of an assessment performed using systems with smart capabilities. The models not only show the elements from different perspectives (specially SAF with its three layers) and their conceptualisations, but also the manner that those components are interrelated.

Although there is an initial validation of the proposed SAF, it is based on an illustrative case study with no real-world implementation. In this sense, there is a necessity to instantiate the elements of the framework in real-world scenarios so as to evaluate its capability to effectively guide the development of smart assessment systems. Therefore, Part II of this thesis is focused on the presentation of approaches that act as concrete implementations of the SAF to solve specific problems.

Part II

Smart Assessment Framework Implementations

A Hybrid Approach to Perform Assessments Using Text Evidence

Contents

4.1	Introduction	57
4.2	Hybrid Approach to Evaluate Business Processes	59
4.2.1	Overview of the Approach	59
4.2.2	LSTM Model	61
4.2.3	Knowledge Base	62
4.3	Case Study	70
4.3.1	Case Study Description	70
4.3.2	Specialisation for Samples Management Process	71
4.3.3	LSTM Training and Testing	72
4.3.4	Assessment Results	73
4.4	Software Tool	75
4.5	The Hybrid Approach in the Context of Design Science	77
4.6	Conclusion	79

4.1 Introduction

Among diverse organisational aspects, business processes are vital, since they encompass a set of activities or functions executed in a specific sequence that provides a desired result to give value to the organisation [46]. Moreover, most of the problems related to efficiency and effectiveness in an organisation are originated in non-mastered processes [22]. Given their relevance, enterprises must understand the state of their processes so as to have awareness of their strengths and weaknesses, and to identify process aspects that require improvement. In this sense, the first step towards process improvement in enterprises is process assessment [254], which allows to have an objective understanding of the processes of an enterprise by providing a qualitative or quantitative view of their state.

Process Capability Assessments [76] are focused on measuring the extent to which a process achieves certain specifications. Usually, capability is measured within the context of maturity assessments, which allows to obtain a full overview of some aspect of an organisation. Capability

assessments, on the other hand, are focused on single processes. In general, to perform maturity or capability assessments, some framework or model is used, which defines how to plan, conduct and define the assessment results. Examples of such models are the Capability Maturity Model Integration (CMMI) [255], the ISO/IEC 15504 [94] and ISO/IEC 330xx [16] standards, the Maturity Model for Enterprise Interoperability (MMEI) [24], the Project Management Maturity Model [81], or the Industry 4.0 Readiness Maturity Model [60]. Since these models group knowledge defining relevant aspects regarding how to perform the assessment of a certain entity, ontologies can be used to describe them formally. Ontologies are formal representations of shared and structured knowledge [256], and they are often used as knowledge base in Artificial Intelligence (AI) applications [257]. Since they are defined through formal data structures, they can be interpreted by machines and further processing can be performed so as to derive new knowledge through diverse inference mechanisms [258]. Since ontologies represent knowledge that is explicitly defined by humans, they are considered in this work as model-driven or classical AI methods (also known as symbolic AI).

A relevant element to perform assessments is the assessment evidence, since it provides a reflection of the As-Is state of the assessed process. Indeed, the evidence is the main source of information used by assessors. Among the various possible types of assessment evidence, a common one is text data, which can contain regulations, strategies, activities descriptions, interviews transcriptions, etc. In this sense, the automatic treatment of text data has risen during the recent years due to the availability of large amounts of text [259], which was possible due to factors such as digitalisation, the rise of social media or the wide use of web and mobile applications that generate text. Other relevant elements that fostered progress in text data processing are the development of new hardware configurations that enhanced the available computing power, and the advances in algorithmic design that allow to learn patterns from data [259]. In this context, a field that has produced massive advances in text data processing, among other applications, is Deep Learning (DL), which encompasses methods that have improved the state-of-the-art in applications such as speech recognition, visual object recognition, or text processing. Since DL models learn directly from data without the necessity of human participation, it is possible to group them into the data-driven AI category.

Considering the advantages of DL and ontologies used for the inference of new knowledge, the use of DL techniques to process text data in combination with ontologies, rules and reasoners to perform activities such as results determination or the definition of improvement suggestions could serve as a valid methodology to support assessments in enterprises. Specifically, the Research Question (RQ) of the work presented in this chapter is related to the assessment of business process capability in enterprises using these techniques to treat text data:

- Is a hybrid Deep Learning and Ontology-based approach valid to perform process capability assessment using text data as evidence?

Given this RQ, the objective of this chapter is to define and evaluate a methodology to automatically perform business process capability assessment using raw text as assessment evidence. To achieve this objective, an approach based on the combination of data-driven AI and model-driven AI methods is defined. Hence, a hybrid approach is devised, aiming at taking advantage of the possibilities provided by both methods to support process capability assessment. The data-driven method used in this work is the Long Short-Term Memory Network (LSTM) [260], a type of Artificial Neural Network [261] that is intended to solve problems in which long-term temporal dependencies must be learned. On the other hand, the model-driven approach is based on the use of an ontology defining aspects of process capability assessment and a set of rules

to infer new knowledge using the existing one [256]. The proposed method is validated through a case study based on performing the assessment of a process comprising activities related to the management of chemical samples in a research institute. Moreover, this chapter presents a software tool developed with the objective to integrate the proposed approach into a web application that can be accessible by end-users to perform assessments. The overall implementation of the approach introduced in this chapter is based on the structure and concepts defined by the Smart Assessment Framework (SAF) introduced in Chapter 3. Therefore, this work also has the meta-objective of evaluating the validity of SAF as a framework to design systems to perform intelligent assessments.

This chapter is organised as follows. Sections 4.2 and 4.3 detail the proposed approach and the case study for validating the approach, respectively. Section 4.4 describes the characteristics of the software tool. Section 4.5 presents details of the development of the hybrid approach and the software tool as research artefacts within the Design Science Research (DSR) context. Section 4.6 presents the conclusion and enumerates future research perspectives.

4.2 Hybrid Approach to Evaluate Business Processes

This section presents the approach introduced in this chapter. First, the instantiation of the SAF to perform business process capability assessment is presented. After, the architecture of the neural network used to treat the assessment evidence is described. Finally, the ontology devised to be used as knowledge base in this work is detailed.

4.2.1 Overview of the Approach

The SAF is instantiated to support Business Process Capability Assessment. In this context, the Assessment Data is composed of text containing a set of descriptions of the assessed process, provided by expert assessors after performing interviews with one or more process responsible. The Assessment Result must define the Capability Level achieved by the process and a set of improvement recommendations that will be further provided to the stakeholders. Figure 4.1 shows the instantiation of the elements of the Application Services (Business) layer of the SAF, which is the focus of this work.

The Assessment Model, an element of the Assessment Process (Business) layer of the SAF, that is used in this work is based on the ISO/IEC 330xx standard [16] series, which provides a schema to be followed by assessors to effectively perform qualitative process assessments. The standard defines two types of Practices: Base Practice and Generic Practice. The Base Practice is an activity that contributes to achieving a specific process purpose, it represents the achievement of the Process Performance attribute of the framework, from the Capability Level 1, and it is specific for the assessed process. Generic Practices are activities that contribute to the achievement of a specified process attribute, and they correspond to practices of Process Attributes from Capability Levels 2 to 5. They are more generic and can be applied to any process regardless their application domain.

In the scope of the implementation introduced in this work, the first step of the assessment pipeline is to consume Assessment Data and provide a rating to each Practice defined by the Assessment Model. The idea is to use the rating values as base for the Capability Level calculation. For this purpose, the Assessment Data (text data) is divided into a set of sentences, with each sentence related to a specific Practice. In Figure 4.1, for instance, evidences ed_{11} and ed_{12} explain the Practice P_1 . More generically, for the Practice P_m , there is a set of n evidences

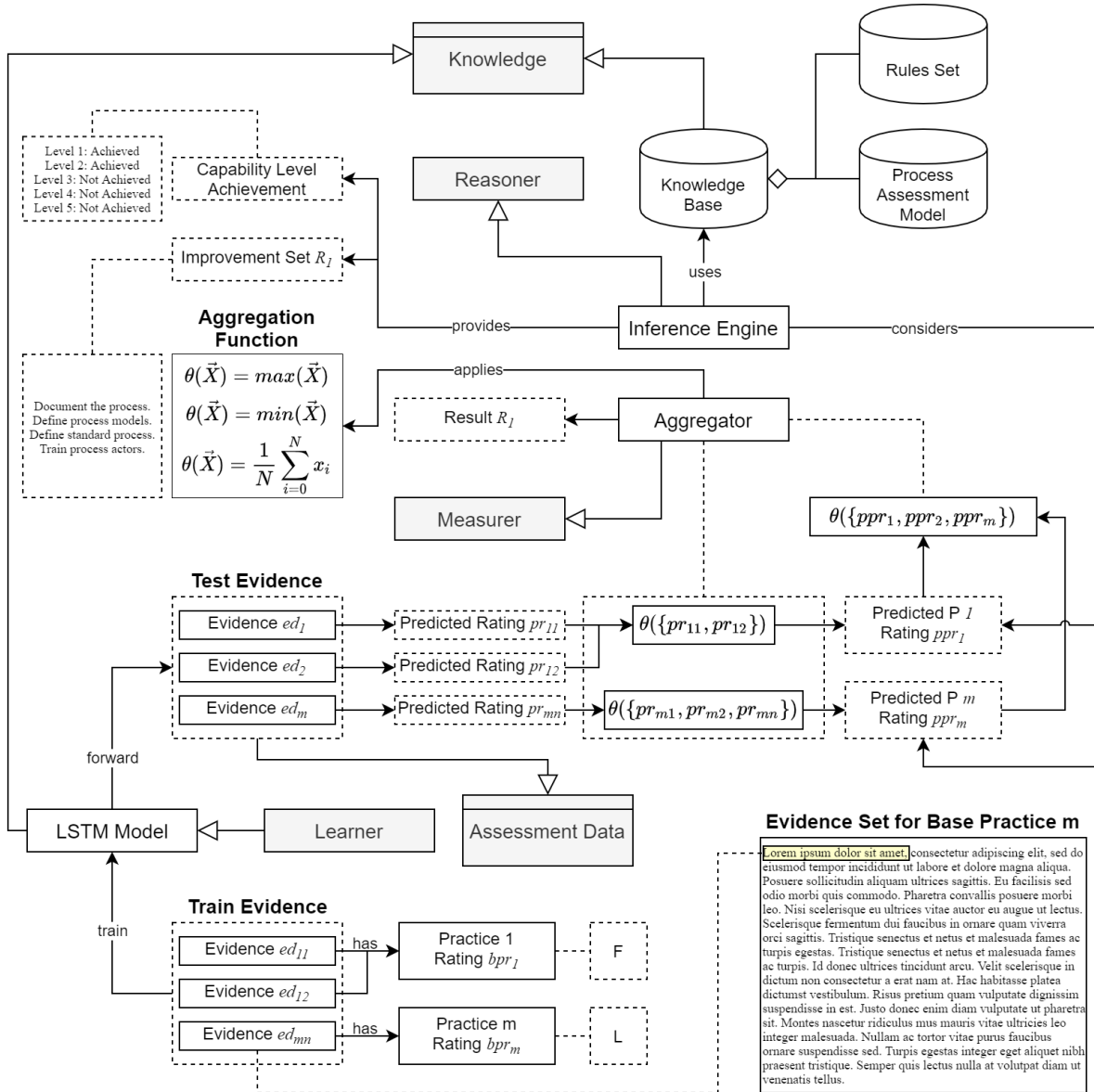


Figure 4.1: Overview of the approach followed in this work.

describing it. Note that this definition is generic enough to be used for any data type. However, for this specific implementation it is based on text sentences associated to a certain Practice.

The problem of automating the calculation of the rating of a Practice using text evidence is wide and can be tackled through different strategies. A suitable approach is to consider it as a classification problem, in which the input of a classifier is the process description and the class is the rating provided by the assessors. This strategy is followed considering a supervised machine learning scenario [262]. The main idea is to train an LSTM network as the classifier using a training dataset and then use the trained model to support the classification, as shown in the bottom-left segment of Figure 4.1. Note that the model and the approach used to train it encompass an instance of the Learner component of SAF. Moreover, the trained model can also be considered as Knowledge within the scope of the framework. The architecture of the LSTM network used in this work is described in Section 4.2.2.

Once the model is trained and it is capable to provide ratings for input sentences, it is able to receive input evidence to produce results. The output of this step of the pipeline is a set of ratings grouped according to the Practice that is associated to the input evidence. These must be then aggregated using a function θ . Figure 4.1 shows an example of this step for the Practice P_1 , in which there are two predicted ratings pr_{11} and pr_{12} , and they are aggregated through $\theta(\vec{X})$ where $\vec{X} = \{pr_{11}, pr_{12}\}$. Three possible ways are defined to obtain the final rating: the maximum value, the minimum value, and the average. For the case study of this work, described in Section 4.3, $\min(\vec{X})$ is used. This task is performed by the Aggregator component, which is an instance of the Measurer element of the SAF.

The ratings for each Practice are passed to the Organiser component of the SAF, which uses a Knowledge Base composed of two elements to define the final assessment results: a Process Assessment Model, which is an ontology defining generic concepts related to the Assessment Process and specific concepts defined by the ISO/IEC 330xx standard series; and a set of Rules to be applied on instances of the ontology, which allows to calculate the capability level of the process and possible improvement suggestions to reach a certain level. However, it is also worth mentioning that the ISO/IEC 33020 Measurement Framework defines only Generic Practices, whilst the Base Practices must be defined by taking into account specific aspects of the assessed process.

The following sections describe the architecture of the LSTM Model that is able to provide the ratings using text sentences as input, and the structure of the Knowledge Base devised to provide the final assessment results.

4.2.2 LSTM Model

The LSTM pipeline is presented in Figure 4.2, which shows an overview of the steps performed for a sample evidence ed_{11} . The first step is to receive as input the sentence and tokenize it [263] to obtain a sequence of n tokens t_i for $i \in n$. Each token is then converted into its index value defined within a dictionary. Since n could vary depending on the length of the sentence, a padding operation is applied so that all input sentences have the same global size M . Therefore, the vector containing the indexes is filled with zeros if $n < M$. The value of M is selected considering the size of the largest sequence of the dataset.

The vector \vec{X} obtained from the previous step is then fed to a neural network that has an embedding layer that allows to convert each word of the input vector (represented by the index) into its GloVe vector representation [264]. GloVe is an unsupervised learning algorithm devised to learn vector space representations of words. It is trained using the aggregated global word-word co-occurrence statistics from a text corpus. In this work, the pre-trained 50-dimensional GloVe vectors trained on a text corpus from Wikipedia¹ is used. This first layer is named GloVe Embeddings, as shown in Figure 4.2. GloVe was chosen since the pre-trained vectors are publicly available² and its use is straightforward with well-known DL frameworks.

The output of the Embedding layer is then fed to an LSTM Layer with 128 units and the default Hyperbolic Tangent (tanh) activation function, followed by a Dense Layer with a Rectifier Linear Unit (ReLU) activation function of size 128, which is then fed to a final Output Layer with a Softmax activation function. Since there are 4 possible ratings for an input sentence (0 - Not Achieved, 1 - Partially Achieved, 2 - Largely Achieved, and 3 - Fully Achieved), the size of the Output Layer is equal to 4. To aid generalisation, Dropout (50%) [265] is applied to the LSTM and Dense layers, whilst L2 Regularization [266] is also used for the Dense Layer. Table

¹wikipedia.org

²<https://nlp.stanford.edu/projects/glove/>

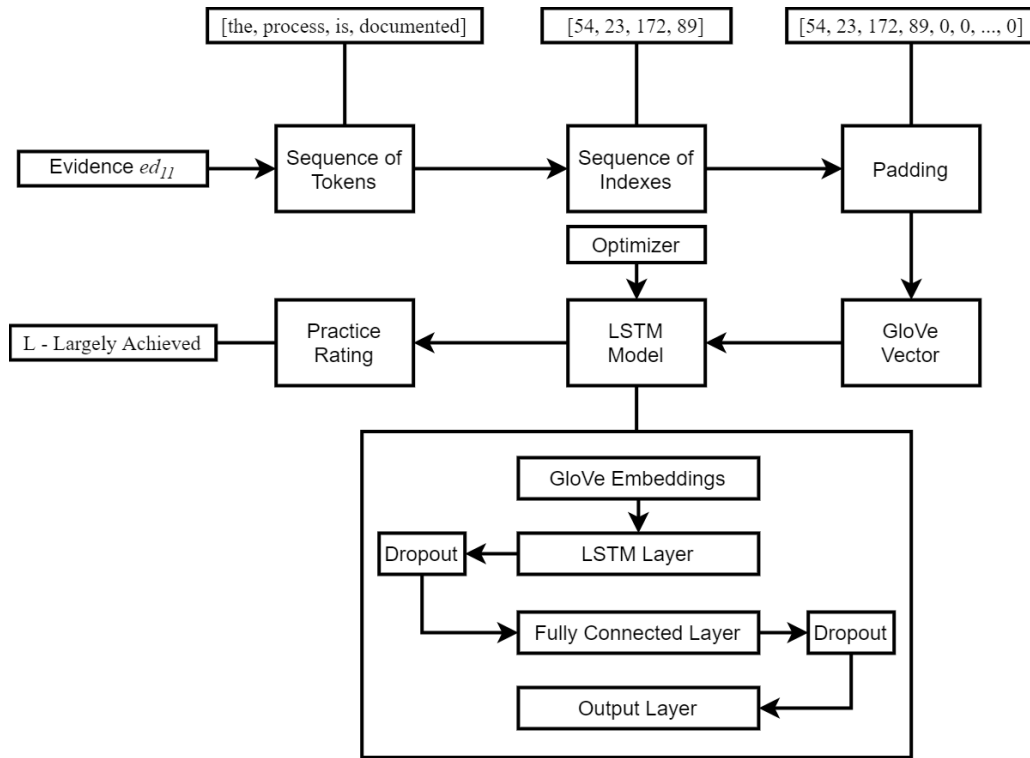


Figure 4.2: Pipeline of the LSTM approach and the architecture of the network.

4.1 shows details of the architecture, which was empirically devised through experimentation with the dataset. It is worth mentioning that the weights of the GloVe Embedding layer are not updated during the training of the network. Hence, it is purely used as a feature extractor and the focus is set on training the layers that are in charge of the classification task.

Layer	Output Shape	Activation	Dropout
Input	[48]	-	-
Embedding	[48, 50]	-	-
LSTM	[128]	tanh	50%
Dense	[128]	ReLU	50%
Output	[4]	softmax	-

Table 4.1: Summary of the architecture of the LSTM network.

4.2.3 Knowledge Base

In the Organisation pipeline of the SAF, the set of ratings obtained through the application of the LSTM Model is passed to the Organiser component that uses a Knowledge Base to provide the final Assessment Result, which is further passed to the Presentation Layer. The Knowledge Base is composed of two main elements: a Process Capability Assessment Ontology (PCAO) and a Rules Set for Process Capability Assessment. Both elements are described in detail in this section.

The Process Capability Assessment Ontology

The Process Capability Assessment Ontology (PCAO) defines a set of classes with attributes and their relationships. The model describes assessment concepts and, more specifically, process capability assessment elements. It is defined using a formal semantic language, this allows to instantiate individuals and apply querying or reasoning techniques on those individuals to derive new knowledge. The idea of using a semantic technology-based PCAO, instead of a simple text file describing process assessment elements, has two purposes: (1) to have a common and shareable understanding of process capability assessment, and (2) to allow the processing of the model using computer programs.

The development process of the PCAO can be explained from the perspective of the three-cycle view of DSR. Within the **Relevance Cycle**, the requirement of the research aims at obtaining a PCAO to support assessments using ratings that are given to qualitative requirements of a process in order to provide a capability level as a result, as well as a set of improvement suggestions. The model must also be defined using semantic technologies so as to serve as knowledge base and support automated assessments.

Within the **Rigour Cycle**, two main works are considered as base of information for the development of the PCAO: the assessment ontology defined by [10] and the ISO/IEC 15502 ontology defined by [258]. The former defines generic concepts from the assessment process such as Requirements, Problem, Solution, and Quality-related aspects following a systemic view, whilst the later provides a meta-model linking concepts that are specific from the ISO/IEC 15504-5 international standard [250]. The work by [258] also introduces an ontology for the Capability Maturity Model Integration (CMMI) [255]. However, this work also focuses on the ISO/IEC 15504-5 standard. Both works pursue the definition of ontologies to support semi-automated assessments. The evolution of ISO/IEC 15504-5 is also considered: the ISO/IEC 330xx series of international standards as knowledge for the design of the model.

The PCAO has two levels: generic level, in which the basic concepts that are related to the assessment process are defined, and the process-specialised level, which defines concepts that are specific for process capability assessment using as core the ISO/IEC 330xx standards. Figure 4.3 presents the first level.

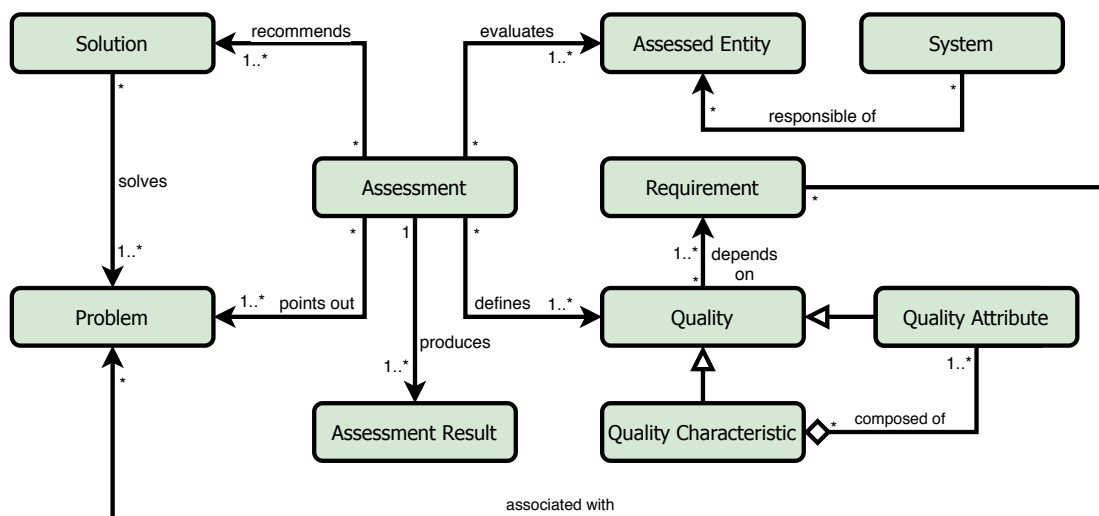


Figure 4.3: The Assessment Ontology.

The ontology defines relationships between basic concepts addressing the assessment process. The main element is the Assessment, its purpose is to evaluate an Assessed Entity, which has a System responsible for it. It also provides an Assessment Result, and it allows to point out a Problem and recommends the Solution that removes it. Moreover, it serves to define the Quality of the Assessed Entity. The Quality is given by a Requirement and there are two types depending on the granularity of its scope: a Quality Characteristic, which defines in a global sense the quality of the entity, and the Quality Attribute, which defines the quality of specific attributes of the entity. The aggregation of several Quality Attributes allows to provide a view of the global Quality Characteristic. Note that the relationships of the model are treated in singular for simplicity although there can also exist one-to-many and many-to-many relationships, as shown in the cardinalities in Figure 4.3.

The second level of the PCAO, presented in Figure 4.4, is a specialisation of the generic level of the Assessment Ontology. It is strongly focused on concepts regarding Process Capability by specialising the existing elements of the generic level to represent process-related concepts.

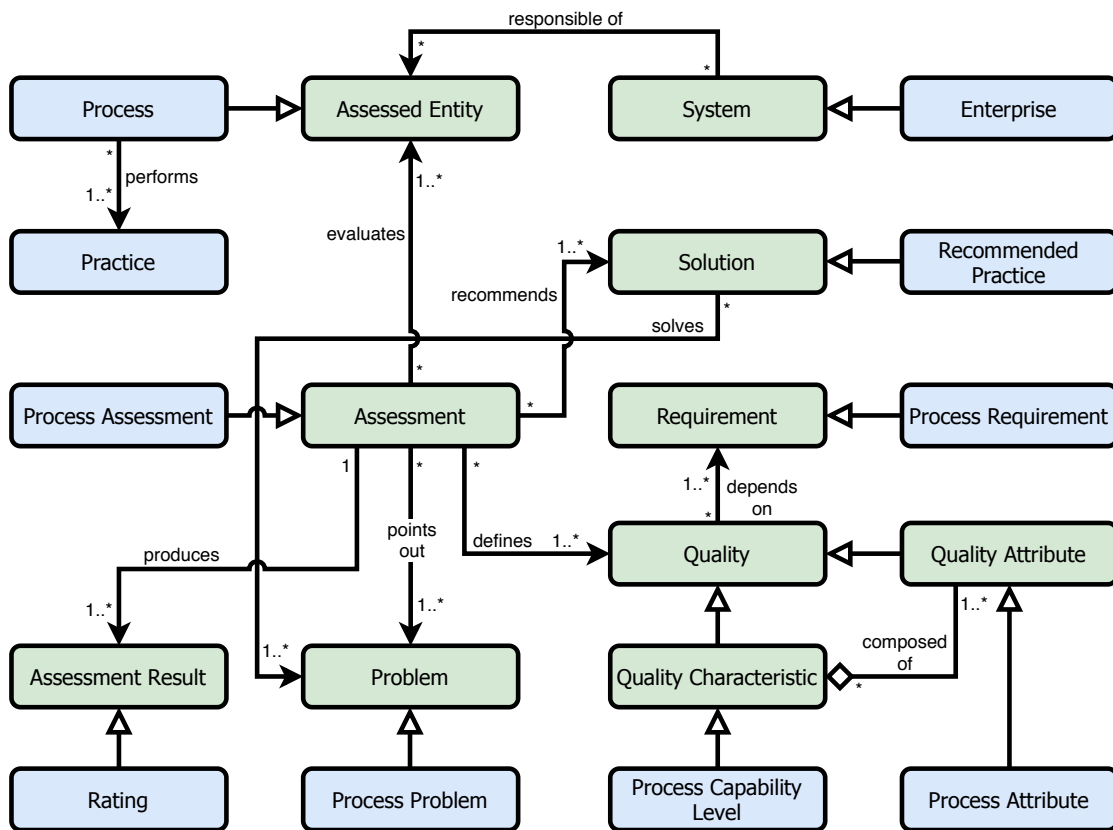


Figure 4.4: The Assessment Ontology specialised for Process Capability Assessment.

In the process-specialised level of the ontology, the Assessment element is a Process Assessment, whilst the Assessed Entity is a Process. Given the context of this work, focused on enterprise assessment, the System responsible for the Process is an Enterprise. Note that a concept described in the ISO/IEC 330xx standard series is introduced: the Practice, which is a type of activity that contributes to the execution of a process [267]. Practices are not generic enough to be considered for the generic level of the meta-model, since they are specific activities that are executed in the context of a process. A Rating

element is also introduced, which is a type of Assessment Result, and the Process Attribute, a Quality Attribute specifically related to processes. The Quality Characteristic in this context is the Process Capability Level. Finally, the Assessment recommends a Recommended Practice as a Solution to a Process Problem. It is worth providing a clear distinction between a Practice and a Recommended Practice. The former is already performed by the assessed Process at the point of time when the assessment is performed, whilst the later is an improvement recommendation that must be implemented by the Enterprise.

For simplicity purposes, only the specialised entities without their relationships are represented in Figure 4.4. However, these design elements are presented in Figure 4.5, in which the elements of the generic level are removed so as to improve visibility of the elements that are only related to process capability. In the ontology, there are inherited relationships from the base elements of the generic level such as Enterprise responsible of the Process (System responsible of the Assessed Entity), or Process Assessment points out a Process Problem, recommends a Recommended Practice, and evaluates the Process. Moreover, new relationships are also introduced such as Process Attribute is achieved by a Practice and it has a Rating. The Process Capability Level is satisfied by the Process Attribute and it can be achieved by a Process. Besides being associated to a Process Problem, in the context of this work, a Process Requirement can also have a Rating.

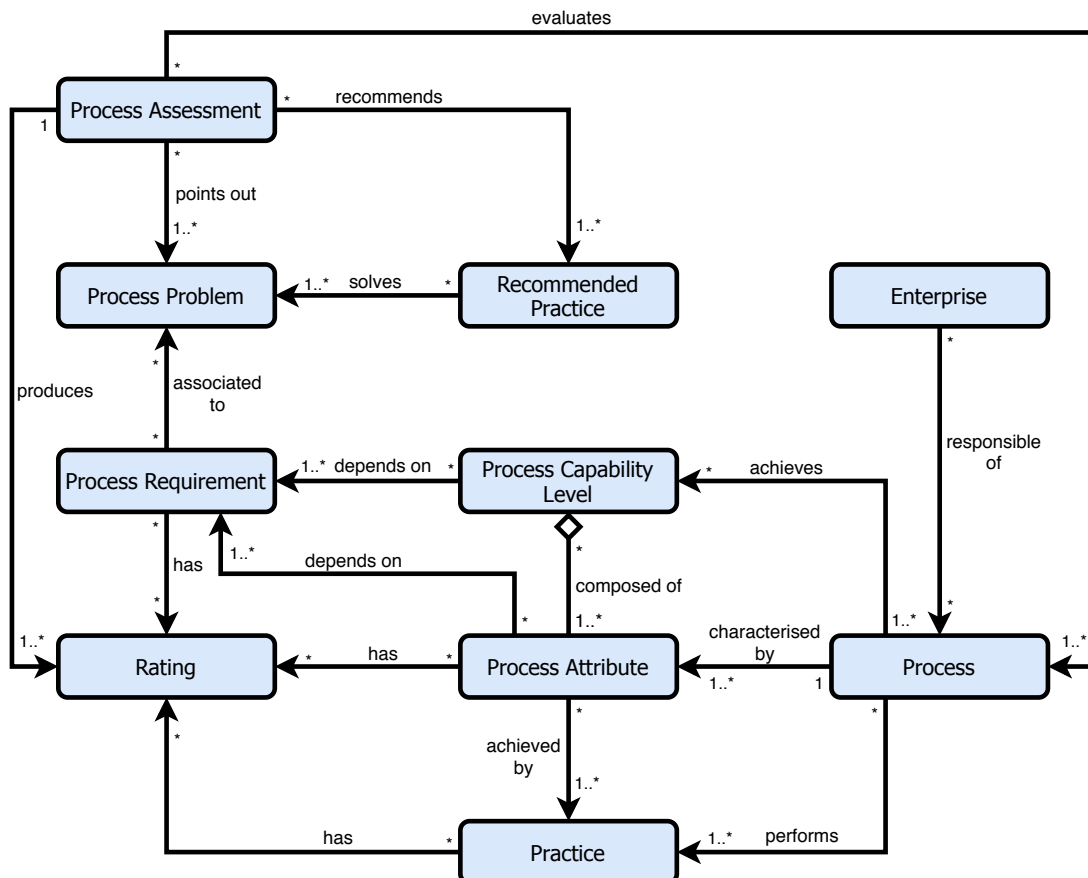


Figure 4.5: The Process Assessment Ontology based on the ISO/IEC 33020 standard in detail.

The ontology described before is represented using semantic technologies. Specifically, the Web Ontology Language (OWL) is used for such purpose. OWL allows to define ontologies using

formal semantics [268] by acting as an extension of the RDF Schema [269], which is a semantic extension of the Resource Description Framework (RDF), a language for representing information about resources in the World Wide Web [270]. In philosophy, ontology is the study of being, of the kinds and structures of objects, properties, events and relations [271]. In the context of semantic technologies, ontologies are formal representations of shared and structured knowledge belonging to one or more domains that are characterised by high semantic expressiveness [256]. Ontologies are commonly used as knowledge bases in AI applications since they allow to validate semantic relationships and derivate conclusions from known facts during inference [256]. They are considered as equivalent to knowledge bases since they are able to represent not only classes and properties but also instances [257].

OWL has been chosen in this work since it is a well-known and widely used language. Moreover, it is part of a standard stack defined by the World Wide Web Consortium (W3C) for the semantic web. The Protege³ ontology editor, version 5.5.0 [272], is used to define the model. Protege is a free open source software that provides a graphical interface to create and edit ontologies. Figure 4.6 presents the main classes of the model at the generic level.

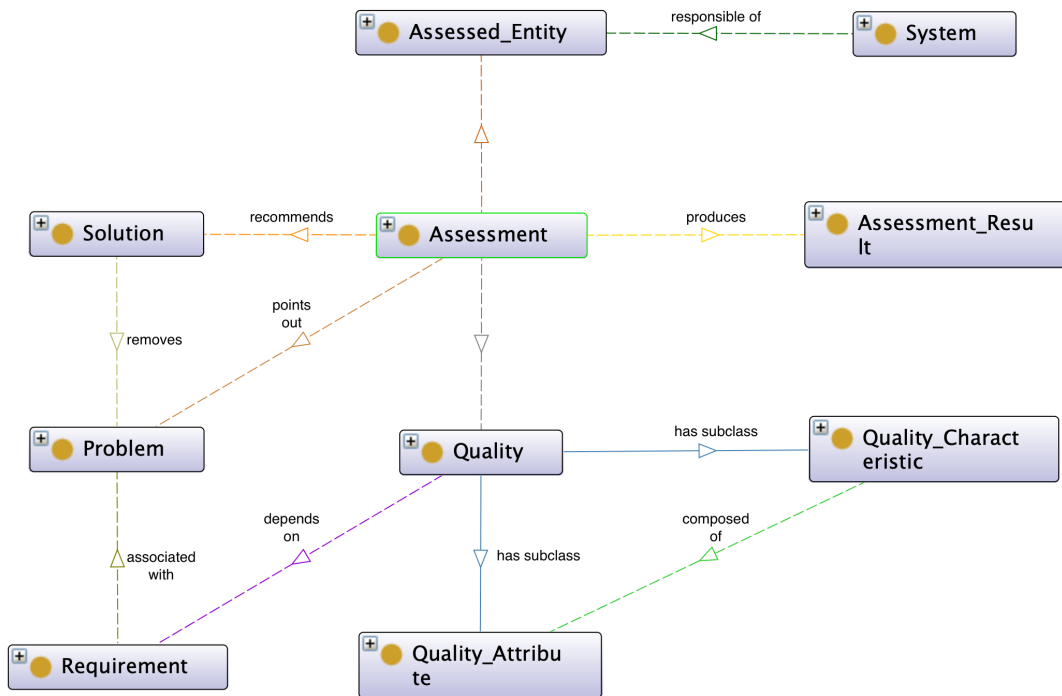


Figure 4.6: Base classes of the ontology used in this work represented through OWL.

All elements of the process-specialised level of the ontology devised using Protege are presented in Figure 4.7. The relationships are not included in the figure for visualisation reasons. However, they are the same as those presented in Figures 4.3 and 4.5.

³<https://protege.stanford.edu/>

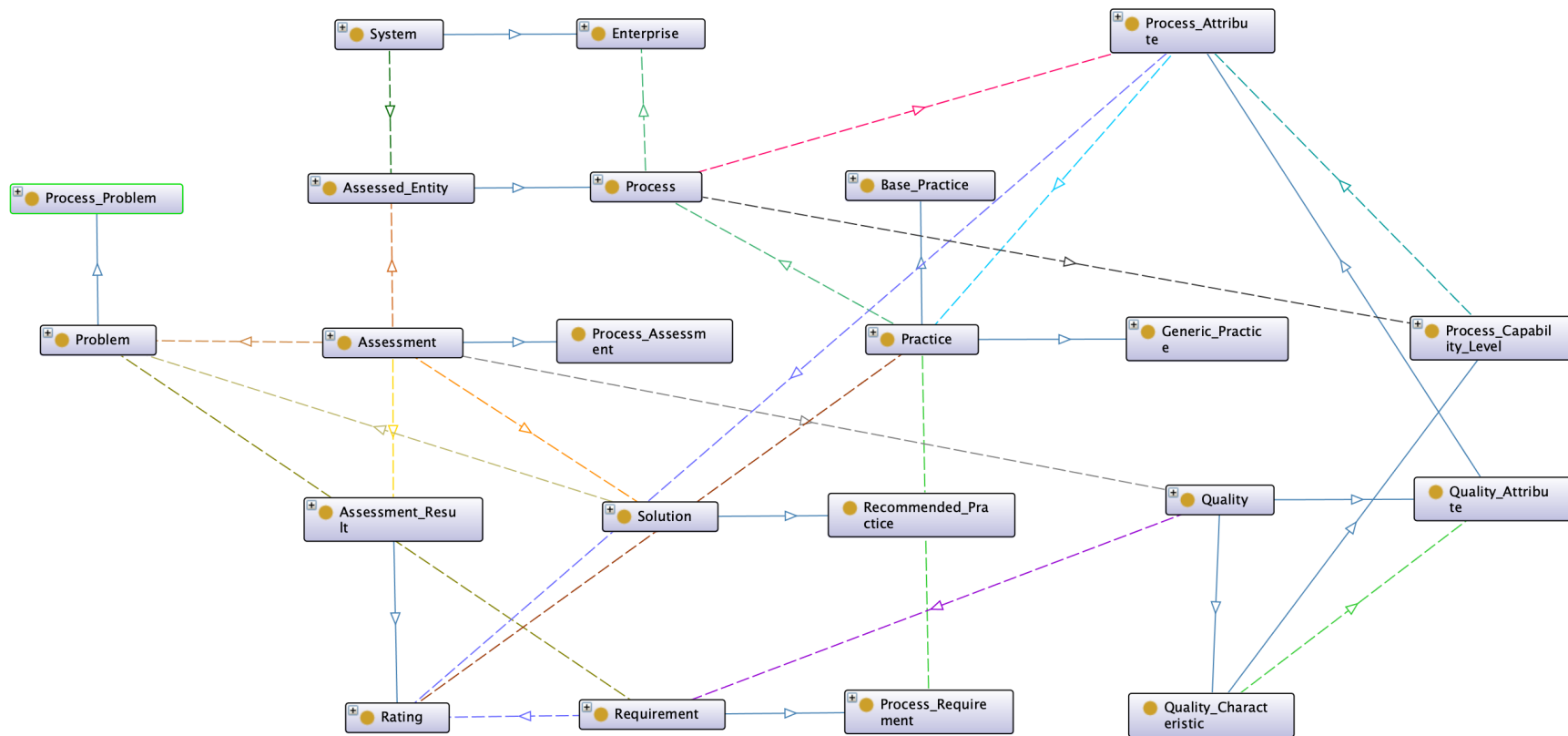


Figure 4.7: Process-specialised classes of the ontology used in this work represented through OWL.

Note that the process-specialised level is still generic in the sense that it is independent from the Assessment Framework used to perform the assessment. An example is the Rating class, a sub-class of Assessment Result. A rating could span different values such as binary (Achieved - Not Achieved), real number (1 to 10 score), percentage of achievement (0 to 100%), etc, depending on the framework that is used. Therefore, a final specialisation must be performed before defining the instances of the classes of the ontology. In this work, this specialisation level is defined using the ISO/IEC 33020 Assessment Framework as core. For this purpose, the Rating class is further specialised to four new classes, following the rating scale provided by the framework: N - Not Achieved, P - Partially Achieved, L - Largely Achieved, and F - Fully Achieved class. The Capability Level class is specialised with five new sub-classes: Capability Level 1 to Capability Level 5. Moreover, the Process Attribute class is specialised with 9 new sub-classes. The Generic Practice class is specialised with 42 new sub-classes. The new sub-classes for both, Process Attribute and Generic Practice, classes are defined in accordance to the elements introduced in the ISO/IEC 33020 standard. As for the Base Practice class, its sub-classes must be defined depending on the characteristics of the process to be assessed. Sub-classes of the Process Problem, Process Requirement, and Process Solution classes are also introduced, considering their link with the sub-classes of the Generic Practices class. An example of a full specialisation and instantiation of the model is presented in Section 4.3.

Figure 4.8 presents the main references that were used to define the elements of each level of the ontology, including the Assessment Framework level, which in this work is the ISO/IEC 33020 Process Measurement Framework standard. The references include the ISO 90001 [251], ISO/IEC 15504 [94], ISO/IEC 33001 [16], ISO/IEC/IEEE 29148 [252], and ISO/IEC 33020 [20] standards. Likewise, other relevant references are the Ontology of Enterprise Interoperability [242], and the General Systems Theory (GST) [239].

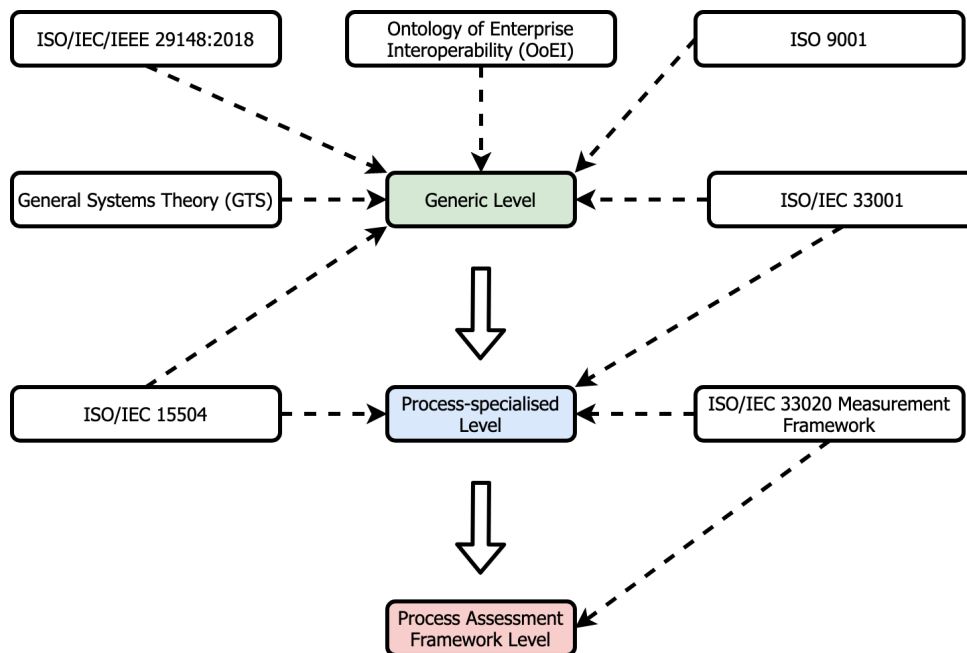


Figure 4.8: References used in each level of the PCAO.

Rules for Process Capability Assessment

The Capability Level calculation is performed through the use of rules that are defined using the Semantic Web Rule Language (SWRL) [273], a language that provides high-level abstract syntax for Horn-like rules to be combined with an OWL knowledge base. Rules defined using SWRL have the form of an implication between an antecedent and consequent [273].

In this work, the main purpose of the definition of a set of rules is to calculate the Capability Level of the assessed process. However, other aspects can also be addressed using rules, such as the definition of possible improvement recommendations given a set of process problems. For each rule, it is first defined in natural language, then it is converted into its equivalent in the human readable syntax defined in the SWRL specification [273], which is a user-friendly syntax to describe rules that are originally defined using the Extended BNF syntax, very similar to the Extended Backus-Naur Form (EBNF) notation used for XML. The ontology editor Protege is used in this work to define and edit the rules, since it provides mechanisms to perform such tasks.

An example of some rules defined in this work is presented next. For the following sentence in natural language: "A *Process* characterised by a the *Process Attribute 1.1 - Process Performance* with a *Rating* equal to *F - Fully Achieved* or *L - Largely Achieved* achieves *Capability Level 1*". Its equivalent OWL rule is defined as follows:

```
Process(?p), PA11_Process_Performance(?pa),
(F_Fully_Achieved or L_Largelly_Achieved)(?r),
Capability_Level_1(?c),
characterisedBy(?p, ?pa),
has(?pa, ?r) -> achieves(?p, ?c)
```

In the case of Process Problems associated with a Process Requirement, the rule in natural language is defined as: "A *Process Requirement GP215RQ4 - Authorities are Defined* that is part of the *Generic Practice GP215 - Define Authorities and Responsibilities*, and has a *Rating* equal to *N - Not Achieved* or *P - Partially Achieved* is associated with the *Process Problem GP215P4 - Authorities are not Defined*". Its equivalent OWL rule is defined as follows:

```
GP215_Define_Authorities_and_Responsibilities(?gp),
GP215RQ4_Authorities_are_Defined(?rq),
GP215P4_Authorities_are_not_Defined(?pr),
(N_Not_Achieved or P_Partially_Achieved)(?r),
composedOf(?gp, ?rq),
has(?rq, ?r) -> associatedWith(?rq, ?pr)
```

Finally, to provide one or more recommendations given the existence of a certain problem the natural language rule is given by: "A *Problem GP215P4 - Authorities are not Defined* that is associated with a *Requirement GP215RQ4 - Authorities are Defined* and that can be removed by the *Recommended Practices GP215P4R1 - Define Process Owner and GP215P4R2 - Define Process Manager*, is removed by the *Recommended Practices GP215P4R1 - Define Process Owner and GP215P4R2 - Define Process Manager*". This statement is defined using the two OWL rules presented next:

```
Rule 1:
GP215RQ4_Authorities_are_Defined(?rq),
GP215P4_Authorities_are_not_Defined(?pr),
GP215P4R1_Define_Process_Owner(?r1),
```

```
associatedWith(?rq, ?pr),
canBeRemovedBy(?pr, ?r1) -> removes(?r1, ?pr)
```

```
Rule 2:
GP215RQ4_Authorities_are_Defined(?rq),
GP215P4_Authorities_are_not_Defined(?pr),
GP215P4R2_Define_Process_Manager(?r2),
associatedWith(?rq, ?pr),
canBeRemovedBy(?pr, ?r2) -> removes(?r2, ?pr)
```

4.3 Case Study

This section first presents a description of the case study that was performed to validate the hybrid approach. A specialisation of the PCAO to support the case study is also described. After, information regarding the training and testing phases of the LSTM network used in this work is described. Finally, the assessment results of the case study are presented.

4.3.1 Case Study Description

To validate the proposed approach, a case study that consisted in performing the assessment of the Samples Management (SM) process of a public research institute was performed. The SM process comprises the activities of reception, treatment, destruction, and returning of different types of chemical samples that are used by the institution. The data describing the process is collected through a series of semi-structured interviews carried out by an assessment team. The interviews were based on asking questions that were guided by an ISO/IEC 15504-330xx [16,250] compliant assessment framework named TIPA [66].

The Assessment Data is composed of Evidence gathered by the assessment team that previously carried out interviews with the process actors. This is the only activity of the assessment pipeline that is performed by humans. The Evidence, registered in raw files by the assessment team, is composed of descriptions from the assessors regarding each Practice of the process. A Rating is also included for each description in the dataset. Hence, the dataset used in this work is composed of a set of instances with the assessor ID, the comment of the assessor, and the ID of the practice. Table 4.2 presents examples of instances of the dataset. Note that the each comment is, in fact, the n -th Evidence ed_{mn} of a Practice m , and it has a Rating pr_m , as presented in Figure 4.1.

Table 4.2: Examples of instances of the dataset.

Assessor	Comment from the assessor	Practice ID	Rating
1	We ask the client or partner by email or orally to give us a list of samples	1	L
1	We should have a little more robustness for encoding: there is no encoding	1	L
2	No high level of risk, rather understanding of expertise	2	F
1	Samples can be destroyed by me, engineers, etc	3	L
2	As far as possible there is a contract to support the activities	3	L

4.3.2 Specialisation for Samples Management Process

To address specific aspects of the Samples Management process, a specialisation of the PCAO described in Section 4.2.3 was devised. In this sense, the class SM Samples Management Process was defined, which is a sub-class of the Process class. On the other hand, a set of four Base Practices that define the Process Attribute 1.1 - Process Performance were instantiated. The Base Practices for the SM process are: BP1 - Management of Samples Reception, BP2 - Management of Samples Processing Operations, BP3 - Return of Samples, and BP4 - Destruction of Samples. Figure 4.9 shows the added Base Practices (highlighted in red) along with the Generic Practices that are defined in the ISO/IEC 33020 standard.

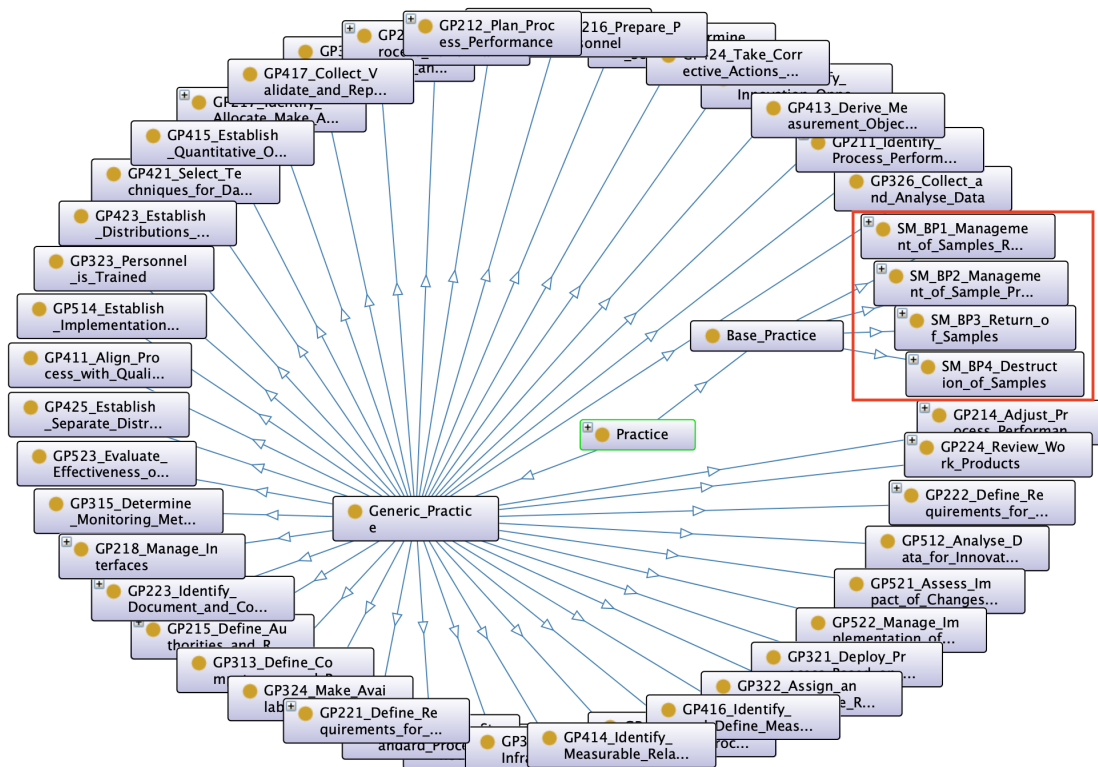


Figure 4.9: Base Practices added for the Samples Management Process, including the already existing Generic Practices.

New rules for reasoning were also defined. Note that Base Practices are specially useful to calculate the *Rating of Process Attribute 1.1*, which is then used to define if the *Capability Level 1* was achieved by the process in the context of ISO/IEC 33020. Considering this, for instance, to define if the *Process Attribute 1.1* has been achieved by the process, it must be checked if all Base Practices have a *Rating* equal to *F - Fully Achieved*. In SWRL, this rule is defined as follows:

```
SM_Samples_Management_Process(?p),
PA11_Process_Performance(?pa),
characterisedBy(?p, ?pa),
SM_BP1_Management_of_Samples_Reception(?bp1),
SM_BP2_Management_of_Sample_Processing_Operations(?bp2),
```

```

SM_BP3_Return_of_Samples(?bp3),
SM_BP4_Destruction_of_Samples(?bp4),
performedBy(?bp1, ?p), performedBy(?bp2, ?p),
performedBy(?bp3, ?p), performedBy(?bp4, ?p),
F_Fully_Achieved(?r),
has(?bp1, ?r), has(?bp2, ?r),
has(?bp3, ?r), has(?bp4, ?r) -> has(?pa, ?r)

```

It is worth mentioning that it is also possible to define sub-classes of the Process Problem, Process Requirement, and Recommended Practice classes that are associated to the new Base Practices. New rules can also be defined to reason using these elements. However, in the scope of the case study, only the implementation of new sub-classes and rules related to the Base Practices was performed since this task already allows to validate the proposed approach.

4.3.3 LSTM Training and Testing

To implement, train and test the LSTM model, a server with the following characteristics was used: Intel Core i9 with 8 cores at 2.3 GHz and 16 GB 2400 MHz DDR4, running macOS Catalina, version 10.15.5. The following Python libraries were used: NumPy, version 1.17.0, Pandas, version 0.25.0, Keras, version 2.3.1, TensorFlow, version 2.0.0, Scikit-learn, version 0.21.3.

Since the interviews for Data Collection aimed at performing Capability Level 1 determination of the process, the dataset described in Section 4.3.1 is composed of Evidence that describes and provides ratings only for the Base Practices of the process. Regardless this limitation in terms of data availability, it can be argued that the proposed approach is easily applicable to assessments that consider higher capability levels, thus including Generic Practices. The dataset is composed of 126 instances. It was shuffled and split to perform hold-out validation: 80% of the dataset was used for training the model (100 instances) whilst the remaining 20% was used for testing it (26 instances).

The LSTM model was trained for 50 epochs, using a batch size equal to 16. To optimise its parameters, the Adam algorithm [274] was used with a learning rate equal to 0.001. The loss function of the network was the cross-entropy loss (L), presented in Equation 4.1, where \hat{y} is the ground truth label, and y is the output of the network for a pattern i considering n patterns fed to the network.

$$L = -\frac{1}{n} \left(\sum_{i=1}^n y_i \cdot \log(\hat{y}_i) \right), \quad (4.1)$$

Note that the size of the dataset is small, thus the model is more likely to overfit [275]. To avoid this issue, regularisers were introduced into the architecture of the model, such as L2 Regularisation and Dropout. At training time, early stopping was also applied, which is based on using a small part of the dataset as validation set, and stopping the training when the performance of the model on the validation set starts decreasing. Since the dataset was very small, the test set was used as validation set. Figure 4.10 presents the loss of the model for the train and test sets during training. Epoch 39 is highlighted in the figure. the best accuracy for the test set was obtained in this epoch, and it was also one of the early epochs in which the test loss started to increase.

Results for the test set of the dataset are presented in Table 4.3. Results show that the worst results were obtained for class P - *Partially Achieved*, for which there is only one sample (support

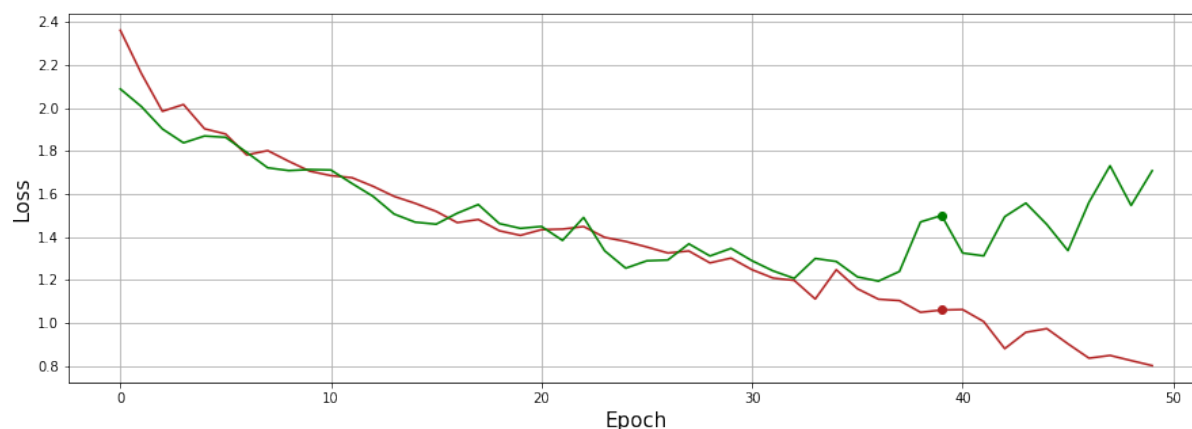


Figure 4.10: Train and test loss during training of the model.

equal to 1). This is likely due to the lack of more available samples in the dataset. Moreover, there is no instance of class *N - Not Achieved* in the dataset, thus there is no result for samples of that class in the table. On the other hand, classification results for the other two classes seem more promising, although the F1-Score for class *L - Largely Achieved* is still low. The easiest class to classify for the classifier is *F - Fully Achieved*, with a F1-Score equal to 0.78, with 1.00 of Recall and an acceptable Precision equal to 0.64. Note that most of the results are very likely due to the imbalance of the dataset and the relatively low amount of training/testing data, which is an issue that could be addressed through techniques such as re-sampling, enhancement of data collection, or using boosting/stacking algorithms. This aspect will be addressed in future work.

Table 4.3: Classification results of the LSTM model.

Class	Precision	Recall	F1-Score	Support
P	0.0	0.0	0.0	1
L	1.00	0.36	0.53	11
F	0.64	1.00	0.78	14

4.3.4 Assessment Results

Once the classification results (predicted ratings) for a set of evidences have been obtained, these must be aggregated using a function θ , as shown in Figure 4.1. Since each rating was given for a sub-comment of an assessor, they were aggregated to provide the final rating for a single Base Practice and Assessor. Note that this is not a definitive aggregation since the ratings of each of the assessors are also required to be aggregated so as to obtain the final rating for the BP. The aggregation function θ used for the case study was $\theta(\vec{X}) = \min(\vec{X})$. Following this aggregation approach, the results for each BP of the SM process are presented in Table 4.4, the real rating provided by the human assessor for the BP is also presented in the table, to serve as baseline for evaluating the automatic approach. Results show a high fidelity to the real rating provided by the human assessor.

The ratings for each BP are passed to the Organiser component of the SAF, which provides a service that applies the pipeline presented in Figure 4.11, which is defined using a Business Process Model and Notation (BPMN) diagram [166]. Some relevant portions of the code such as

Table 4.4: Ratings of each Base Practice after aggregation of the ratings provided by the LSTM Model.

Base Practice	Predicted Rating	Real Rating
BP1	L	L
BP2	F	F
BP3	L	L
BP4	F	F

functions, variable definitions, etc., are also presented in the figure. The process starts by loading the PCAO (a ".owl" file). After, the individuals of the ontology are loaded into local variables and individuals that are not within the ontology are instantiated. Then, relevant relations between the individuals are defined. After, the ratings (previously saved in a ".json" file) defined by the LSTM Model are loaded, and the individuals (Base Practices 1 to 4) receive those ratings. Finally, the reasoner is applied. In the case study, the reasoner that was used was Pellet [276]. With the aid of this reasoner, the rating for the Process Attribute 1.1 was obtained. It is also inferred, using the rating of the process attribute, that the SM process achieves Capability Level 1. Specifically for this implementation, the Python library Owlready2, version 0.24, was used.

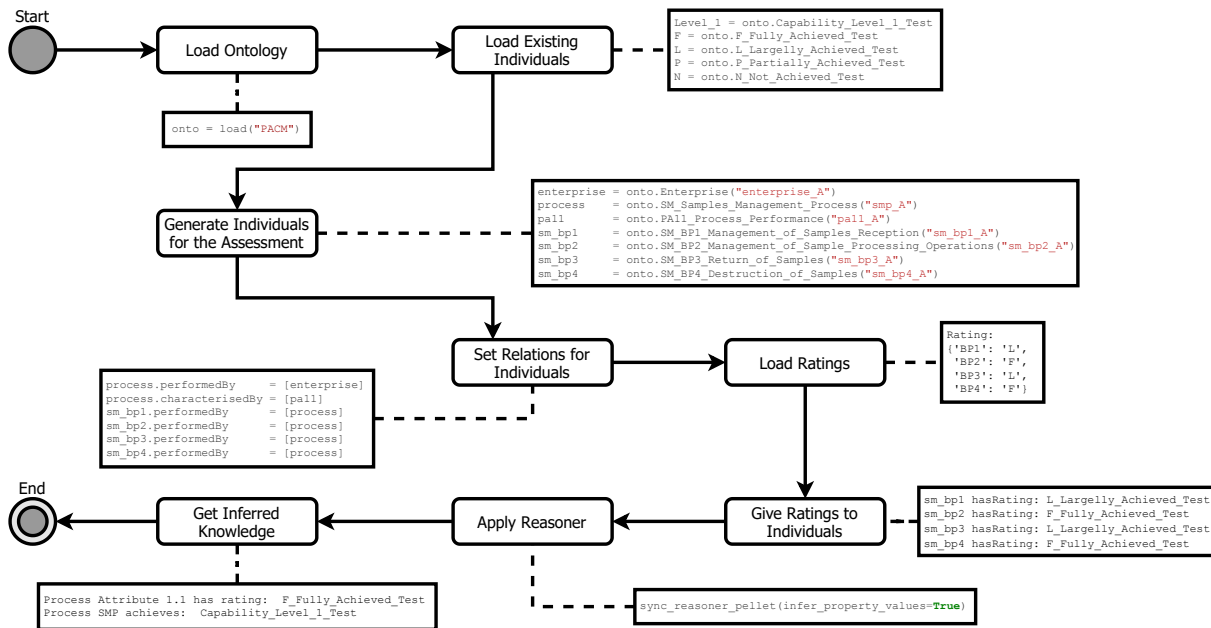


Figure 4.11: Organiser component pipeline and the results for the SM process.

The results presented in this section show the suitability of the proposed approach to perform capability assessment of a business process. The method is able to input text evidence related to certain process practices, and it provides ratings (based on previous assessments made by human assessors) that are used to create instances of an ontology to define results for Process Attribute ratings and Process Capability. Moreover, improvement recommendations can also be provided as assessment result. Note that this method has a main limitation: if the rating provided by the LSTM model does not accurately reflect the reality, this error will be propagated to the final result of the assessment. To tackle this barrier, the model must be efficiently trained and its predictions must be accurate. Likewise, aggregation methods such as the one used in

this work allow to reduce risk of errors in the assessment result. Finally, future work will focus efforts to achieve explainability of the results provided by the model. Explainability is relevant for stakeholders to understand the reasons why a certain rating has been given by the model so as to serve as guide to perform improvement initiatives.

4.4 Software Tool

A software tool has been devised to integrate the approach introduced in this chapter into a web application. The tool was designed as an instantiation of the SAF introduced in Chapter 3. The tool is intended to provide the three main services: Data Perception, Organisation, and Presentation, using the artefacts that were presented in the previous sections of this chapter: the Neural Network (LSTM) and the Ontology (PCAO). The architecture of the software tool is presented in Figure 4.12, which also shows the inclusion of the Knowledge element in addition to the three main services of SAF. This section presents details of the architecture of the system considering how each service is provided and the methods and technologies that are used.

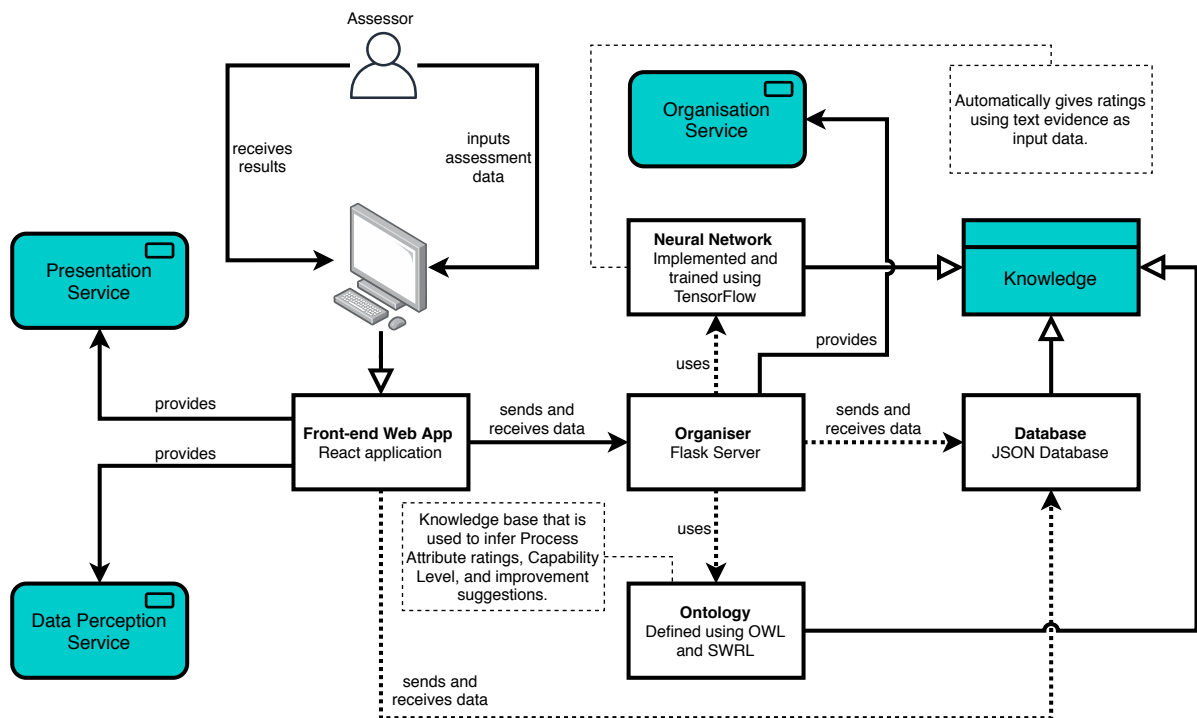


Figure 4.12: Architecture of the system. Elements represented with white boxes are concrete components providing Application Services and accessing Data Objects, whilst the light dotted lines and boxes are used to add information regarding a specific component. Arrows with white triangles represent inheritance.

The Data Perception and Presentation services are both provided through the Front-End part of the tool. In this sense, the input Assessment Data is provided by human assessors directly to the software and the Assessment Results are also given by the tool through a visual interface that can also allow to download them. The version of the tool described in this work provides support for the input of text evidence and it provides the Assessment Result in two formats: text and ontology. The Front-End part of the application was developed using the React framework [277],

which is an open source JavaScript library for building user interfaces. React was developed by Facebook and it was released in 2013. In addition to React, other JavaScript libraries to support it were used such as React Redux⁴, or React Redux Thunk⁵.

The Organisation service is provided by a Back-End application that acts as a server using the Python Flask library⁶. The application receives POST requests⁷ from the Front-End application and it follows the pipeline described in Figure 4.13. The requests contain in their body all data from the assessment including name of the enterprise, name of the assessed entity, evidence type, and the proper assessment evidence.

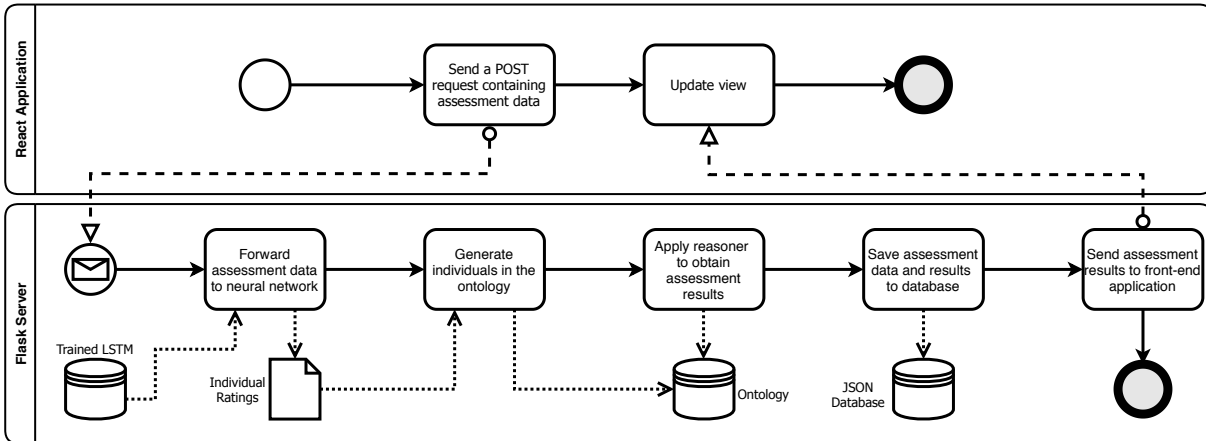


Figure 4.13: Overview of the tasks performed by the Front-End and the Back-End applications of the software tool.

After the reception of the request from the React application, the evidence is formatted and used as input for the LSTM. The network receives as input a set of text sentences describing the process and it provides ratings for each sentence, which are then aggregated to provide the overall rating for a certain Indicator of the process. According to the ISO/IEC 33020 standard [20], Indicators are linked to sources of evidences and explain the value of the attribute of a process. Indicators are organised considering Base Practices and Generic Practices that are grouped into PAs. Note that an aggregation step must be performed in order to obtain the final rating of the PA.

Once the ratings for the PAs that are under consideration are obtained, the next step is to define a new set of individual instances from the PCAO. The PCAO, defines a set of concepts related to Process Capability assessment from the perspective of the ISO/IEC 33020 standard. As shown in Section 4.2.3, the ontology has concepts such as Requirement, Capability Level, Process Attribute, Base Practice, Generic Practice, Recommended Practice, etc. The ratings provided by the Organiser are used to calculate the ratings for Process Attributes and the achievement of Capability Levels. Moreover, recommended practices to improve the assessed process towards the achievement of a certain Capability Level are also defined. These results are then saved in a JSON database through another POST request, and a final answer containing the Assessment Results are sent to the Front-End application. Once the answer from the Organiser is received, the Front-End application updates the view so as to present the results to the user.

⁴react-redux.js.org

⁵github.com/reduxjs/redux-thunk

⁶flask.palletsprojects.com

⁷w3.org/Protocols/rfc2616/rfc2616-sec9

Note that both, the neural network and the ontology could be structured and defined differently depending on the entity to be assessed and the assessment framework that is used. Moreover, the type of assessment evidence is also relevant for the configuration of both elements. The current version of the tool allows to perform Process Capability assessment for text evidence only, and the network and the ontology introduced in the previous sections, are specific for such purpose. However, future work could extend the capabilities of the tool to address this issue. The main screens that are available to the user when an assessment is performed are presented in Figure 4.14.

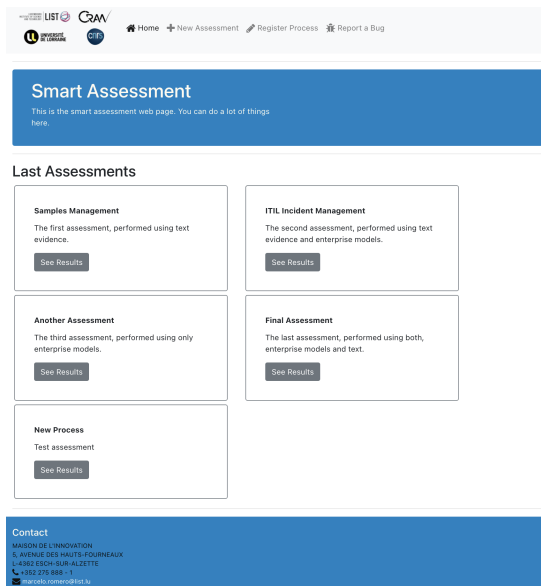
The tool has a home page from which it is possible to use all the features from the tool, this page is shown in Figure 4.14a. The focus is put on the New Assessment screen, since all activities to perform an appraisal are performed in that section of the tool. In the New Assessment screen, the first step is to input general information about the assessment, this is done in the screen presented in Figure 4.14b. Following, the assessment scope is defined, as shown in Figure 4.14c. The text evidence describing the practices to evaluate are then introduced by the user (Fig. 4.14d). Finally, the user clicks on the *Perform Assessment* button, and the process shown in Figure 4.13 takes place. Once the assessment is finished, the results are displayed in a dedicated page (Figure 4.14e).

The last screen presented in Figure 4.14 shows the results for the same case study presented in Section 4.3, although it is now supported by the tool. The assessed entity is a business process that comprises activities of reception, treatment, destruction, and returning of different types of chemical samples that are used in an organisation. The data is configured in text format, and it comprises descriptions of different aspects of the process. The objective is to provide the measurement of the capability level of the considering practices up to capability level 1. Since the objective is to evaluate capability level 1, four specific Base Practices (BPs) were defined to be rated by the system: BP1 - Management of samples reception, BP2 - Management of samples processing operation, BP3 - Return of samples, and BP4 - Destruction of samples. The system provides ratings for each practice and they are then aggregated to define the final rating for Process Attribute 1.1 - Process performance (PA 1.1). If the rating of PA 1.1 is equal or higher than L - Largely Achieved, then capability level 1 is reached. Note that possible improvement recommendations for reaching capability level 2 were manually added for illustration purposes, these were included only to show that, with the proper amount of data, the tool can provide a larger number of improvement recommendations depending on the achievement (or not) of a capability level and on the ratings obtained for the Practices of the process.

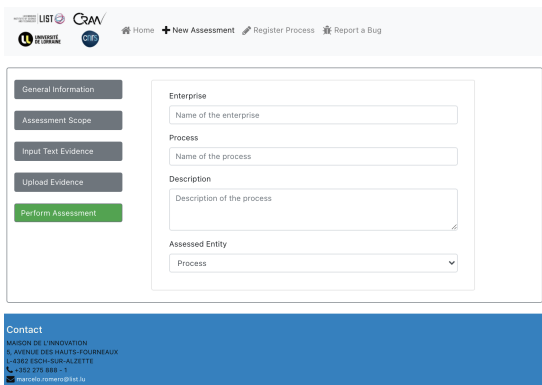
4.5 The Hybrid Approach in the Context of Design Science

From the perspective of the DSR methodology [48], this chapter introduced two artefacts: the hybrid approach for process capability assessment using text evidence and the software tool to use an implementation of the hybrid approach to aid end-users to perform assessments.

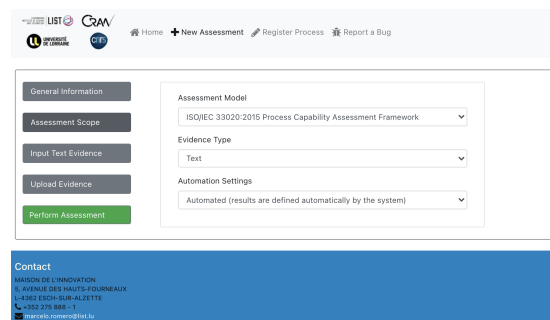
For the **Hybrid Ontology and DL approach** to treat text evidence, presented in Section 4.2, the **Relevance Cycle** considers as requirement the implementation of a system able to take advantage of data-driven and model-driven AI methods to support the automatic treatment of text evidence in order to perform business process capability assessment. This has been accomplished within the **Design Cycle** that was focused on building a knowledge base with the structure of an ontology, the rules for inferring assessment results, and the architecture of the neural network to process the text data. As main sources of knowledge within the **Rigour Cycle**, the following elements have been taken into consideration: literature addressing OWL ontologies,



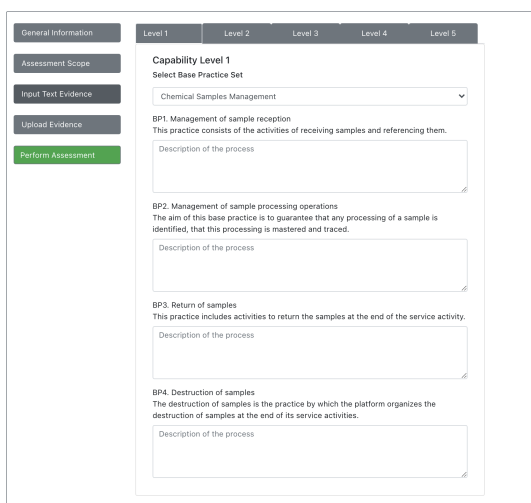
(a) Home page of the software tool.



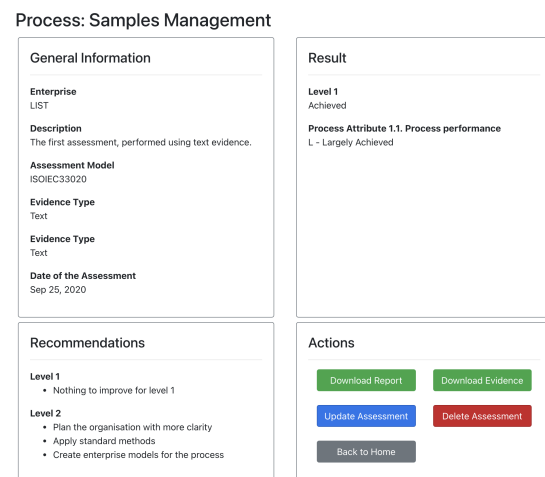
(b) Screen to input general information of the assessment.



(c) Screen to define the assessment scope.



(d) Screen to input assessment evidence.



(e) Screen that shows the assessment results.

Figure 4.14: Screens of the tool that are used during an assessment.

SWRL documentation and literature, the standards and literature addressing process capability assessment, and courses, tutorials, documentation and papers addressing the implementation and testing of neural networks focused on natural language processing.

As for the **Software Tool**, presented in Section 4.4, the requirement within the **Relevance Cycle** was a web application able to support automated assessments that receive text evidence as input from the users and present the results. Therefore, the main idea of the development of this tool was to provide Data Perception and Results Presentation services using the hybrid approach as core to perform an assessment. The **Design Cycle** was based on defining the architecture of the system and implement it using the available development resources and tools. The selection of the languages and frameworks used to develop the tool was done considering: the components of the SAF, the experience and expertise of the developer, and the amount of open resources that were available online to solve problems or clarify doubts that could appear during the development process. Likewise, this development expertise and the documentation addressing web development, servers configuration, database configuration, and DL frameworks were considered as the knowledge base within the **Rigour Cycle**.

4.6 Conclusion

This chapter introduced an approach to perform Process Capability Assessment in enterprises guided by the concepts defined by the SAF. The approach is based on the use of an LSTM, a well-known type of neural network in the DL field, to treat input data in text format combined with a knowledge base and an Inference Engine to provide the final assessment results. Research efforts in this work have been focused on reaching a suitable LSTM architecture and training strategy to tackle the problem. The efforts were also focused on the development of the PCAO, a process capability assessment ontology, and the proper rules to be used as knowledge base to provide Capability Level calculation capability. The PCAO is strongly based on the ISO/IEC 330xx standard series and other well-known sources of information. Moreover, this chapter also introduced a software tool devised to integrate this hybrid approach into a web application to allow users to perform semi-automatic assessments. A case study was performed to validate the approach and answer the RQ. It was focused on performing capability assessment of a chemical Samples Management process. For the case study, the PCAO and the rules of the knowledge base were adapted to consider specific aspects of the process. Moreover, the LSTM model was trained in such a way that it was able to provide ratings to practices of that process. Results of the case study showed the suitability of the proposed approach. Likewise, the functioning of the software tool was presented using the same case study.

Future research efforts could be focused on validating the SAF through other methodologies besides the one presented in this chapter. Moreover, new case studies to evaluate other organisational aspects such as enterprise interoperability, risk management, digital transformation readiness, etc., could be performed. Likewise, since the PCAO is intended to be compliant with the ISO/IEC 330xx standard series and it is strongly focused on process assessment, future work could aim at considering other assessment frameworks and standards to treat different enterprise entities such as the Capability Maturity Model Integration (CMMI) [255], Maturity Model for Enterprise Interoperability (MMEI) [24], or the Project Management Maturity Model [81]. Finally, regarding the assessment evidence, it could differ from plain text data, thus the treatment of other enterprise information sources such as sensor data, regulatory documents, multimedia, etc., could be considered. Depending on the type of data to be treated, the treatment method that is used will differ. In this work, a natural language processing technique was used to treat

text evidence, but other methods could be considered depending on the available data. In this sense, the use of enterprise models as assessment evidences is explored in the following chapters of this thesis. Specifically, the work presented in Chapter 5 serves as initial ground to support the development of approaches and tools for automating the process of the analysis of enterprise models, since it introduces a framework to define the assessment requirements that can be answered or evaluated using models designed with a certain modelling language. On the other hand, the automatic treatment of enterprise models (specifically process models) to provide assessment results from a data-driven perspective is addressed in Chapter 6.

Evaluating the Suitability of Enterprise Modelling Languages to Support Assessments

Contents

5.1	Introduction	81
5.2	Enterprise Models and Assessments	83
5.3	Modelling Language and Assessment Requirements Analysis	84
5.3.1	Requirements Decomposition	86
5.3.2	Requirement and modelling elements matching	89
5.4	Analysing BPMN and ArchiMate Considering ISO/IEC 33020	90
5.4.1	Considerations	90
5.4.2	Process Attributes and Indicators of ISO/IEC 33020	91
5.4.3	Results and Discussion	92
5.5	Assessing Capability Level of a Business Process	95
5.5.1	BPMN	97
5.5.2	ArchiMate	102
5.5.3	Discussion	105
5.6	The Enterprise Models Analysis Approach in the Design Science Context	106
5.7	Conclusion	106

5.1 Introduction

In the Data Collection phase of a Business Process Maturity or Capability Assessment, there are several enterprise elements that could serve as evidence such as documents, e-mails, interview extracts, questionnaires, or enterprise models. The latter provides a representation of organisational aspects [102] such as structure, information, resources, goals, etc., and they can be used for different purposes such as understanding and documenting the organisation, planning the management of change, designing and re-engineering enterprise systems, among others [47]. In this context, an enterprise could have one or more models that could be used as basis to perform

appraisals. However, the analysis of enterprise models is time-demanding, specially if it is performed by human assessors. Therefore, reducing the amount of necessary effort to perform such analysis could be valuable to improve the overall assessment process. In this sense, a possible manner of improving this process is to understand beforehand which type of information can be found within an enterprise model. To reach this understanding, two main elements must be taken into account: the modelling language that has been used to represent the model, and the list of requirements defined by the assessment framework that is used to perform the assessment. To tackle this issue, this chapter aims at defining a methodology to identify the requirements, devised by an assessment framework, that can be answered through the analysis of modelling elements defined using a specific modelling language. In this sense, the work presented in this chapter is focused on answering the following Research Question:

- Can enterprise assessment requirements that are able to be answered by the elements of a modelling language be identified through a specific methodology?

The proposed approach is intended to aid assessors at identifying, through a simple process, the requirements that can be answered using models, available for organisational assessment, by taking into consideration the modelling language in which the model was defined. Specifically, this chapter introduces a structured approach based on a requirements decomposition method [278], and matching approach, able to link assessment requirements with modelling elements, based on the Goal Question Metric (GQM) paradigm [279]. It is expected that this methodology will enable assessors to reduce time when reviewing evidences during appraisals, allowing them to consider only those requirements that are visible through the enterprise model that is being analysed. Note that the proposed methodology could also allow to optimise the time required for the development of a system in charge of automating appraisals. For instance, given that the objective is the development of a system able to automatically input and process enterprise models, such development could be directly centred on designing and implementing features that are based on evaluating only the requirements that were previously isolated following the application of the proposed approach. In this context, such approach can be viewed as a previous step of the development of smart assessment systems in the context of the Smart Assessment Framework (SAF).

To validate the approach, a proof of concept based on the analysis of the capability of Business Process Modelling Notation (BPMN) [166] and ArchiMate [237] languages to support requirements defined by the ISO/IEC 33020 international standard [20], an assessment framework devised for process capability assessment, is presented in this work. Moreover, a case study is introduced with the objective to show that the proposed approach can also be used to analyse concrete enterprise models to provide ratings to the attributes of a specific business process.

The contributions of the work presented in this chapter are manifold: (1) a methodology to identify the requirements from an assessment framework that can be answered through the analysis of modelling elements defined by a specific modelling language, (2) the implementation of the methodology to evaluate which requirements (defined in terms of results of achievement of a Process Attribute) of the ISO/IEC 33020 can be answered by the BPMN and ArchiMate languages, (3) a description of the possibilities of BPMN and ArchiMate to provide assessment results following the ISO/IEC 33020 measurement framework for assessment of process capability, (4) the assessment of a real-world business process through the analysis of enterprise models describing its characteristics.

This chapter is organised as follows. Section 5.2 briefly describes how enterprise models are treated within the context of enterprise assessment in the recent literature. The proposed

approach is then introduced, in Section 5.3, describing the process to perform a modelling language analysis for answering requirements defined by an enterprise assessment framework. This methodology requires validation, thus Section 5.4 describes its application for the evaluation of specific modelling languages (BPMN and ArchiMate) considering a certain assessment framework (ISO/IEC 33020). Moreover, since the approach can also be used as basis to directly perform assessments, Section 5.5 presents a case study that consists of analysing two enterprise models to obtain assessment results. On the other hand, since the proposed approach is devised as an artefact from a Design Science Research (DSR) perspective, Section 5.6 details how the approach is positioned in this context. Finally, Section 5.7 enumerates the conclusions from the work and introduces future research perspectives.

5.2 Enterprise Models and Assessments

The relationship between enterprise models and the assessment process has been addressed in the literature during the recent years. Specifically, enterprise models are often used as base to provide an evaluation of the state of certain organisational aspect through some assessment methodology. The work by [93], for instance, presented an approach to perform maturity assessment of information governance through the analysis of enterprise architecture models and descriptive logic. The approach was based on using ontologies to represent enterprise architecture models of organisations through a converter, and executing descriptive logic queries obtaining results that were analysed afterwards to define the assessment results. In [107], an ontology-based privacy compliance checking approach to detect possible privacy violations in clinical workflows was proposed. The Privacy-aware Clinical Workflow (PaCW) Ontology was introduced in the work. It is based on the Privacy Ontology and the BPMN Ontology, and it is extended with principles based on the General Data Protection Regulation (GDPR)⁸. The paper also presents a tool that allows inferring using BPMN models and checking if it is compliant with the GDPR regulation. On the other hand, [108] introduced a methodology that relies on dependency analysis to identify entities that are dependent on other entities to assess the impact of the changes in business process models. For this purpose, an ontology describing dependency relationships was defined. The model was devised to serve designers and business experts to estimate risks associated with changes in processes along with the inherent implementation efforts. The work uses the ontology defined in [280], which contains definitions of BPMN concepts and it was extended to contain dependency relationships. In [109], the authors proposed an enterprise architecture meta-model of ISO 27001 [281] and its mapping to COBIT 5⁹ to be used as a complement of textual representations. The work aimed to facilitate the COBIT 5 and ISO 27001 simultaneous assessment, with the objective to evaluate Enterprise Governance of Information Technologies. Finally, the work by [110] explores BPMN diagrams and Business Rules (BR) to be used for Knowledge Management. The framework presented in the work is based on the combination of BPMN and BR as a tool for Knowledge Management and it considers specifically the formal knowledge and not the internal knowledge from the individuals of an organisation. The framework is not intended for concrete assessment activities of business processes but it focuses on providing formal knowledge for Knowledge Management that could be useful for assessment activities.

The research work described before intended to support the assessment process using enterprise models as assessment evidence. In some cases, ontologies and rule-based systems are used to support a more automated assessments. Likewise, some works have also aimed at eval-

⁸<https://gdpr.eu/>

⁹<https://www.isaca.org/resources/cobit>

uating modelling languages or techniques that are able to automatically provide models using some kind of input data. In [282], for instance, the potential of Process Mining [283] to support software process assessment and improvement approaches has been addressed. Generic Practices of the Capability Maturity Model Integration [18] has been analysed by the authors, with the objective to define which Generic Practices can be answered with models obtained by applying Process Mining techniques on process logs. The analysis, in this case, was more focused on the Process Mining methods rather than on enterprise models or modelling languages. On the other hand, [284] defined the notion of Multi-View Modeling (MVM) capability and proposed a formalised description based on Extended Backus Naur Form (EBNF) to be used as basis for assessing the MVM capability of a modelling method and identifying requirements to achieve it.

Given the works introduced in this section, there seems to exist a lack of research work that specifically deals with the evaluation of the elements defined in enterprise models to answer the requirements defined by an assessment framework in order to perform enterprise assessment. In this context, enterprise models are used as evidences during an assessment to verify to which extent an assessed entity is mature. As a first step to perform the assessment, it is necessary to understand if the modelling language used to represent the available enterprise model provides the elements that allow to verify a set of requirements that will be studied. The second step implies the proper verification of the enterprise model, considering only those requirements that can, indeed, be checked through the analysis of the model. The work presented in this chapter deals with these steps, proposing a structured approach based on a requirements decomposition method [278], and a requirement and modelling elements matching approach based on the Goal Question Metric (GQM) paradigm [279]. It is worth mentioning that the main contribution of this work is the evaluation pipeline introduced in the next section. The approach is intended to be generic enough to serve as a mechanism that can be used for the analysis of different modelling languages and assessment frameworks. In this work, the ISO/IEC 33020 assessment framework and the BPMN and ArchiMate languages are used as objects of analysis to test the applicability of the proposed approach. Moreover, the result of this analysis can also be viewed as a proper contribution, since it is composed of a list of atomic requirements that can be answered by these languages. The list can be used for future assessments to target only those requirements that are able to be "seen" by inspecting models that were designed using those BPMN and ArchiMate. Regarding the Smart Assessment Framework (SAF), the smart systems defined using such framework could be focused on treating only the requirements that are "visible", in the case that enterprise models are going to be considered as assessment evidences.

5.3 Modelling Language and Assessment Requirements Analysis

This chapter introduces a method devised to define the linkage between two elements: assessment frameworks and enterprise models. The main objective is to provide means to identify if the requirements defined within the scope of an assessment framework can be answered by the modelling elements of enterprise models defined through the use of some modelling language. This will allow assessors to understand *a priori* which enterprise models are relevant to check the requirements defined by the assessment framework taking into consideration the modelling language used to represent them, thus improving the verification step of the available models during an assessment. Figure 5.1 presents a meta-model describing an overview of the aspects that must be considered for the analysis of both, enterprise models and modelling languages. We use the Unified Modelling Language (UML) diagram [234] to visually represent the elements of the meta-model.

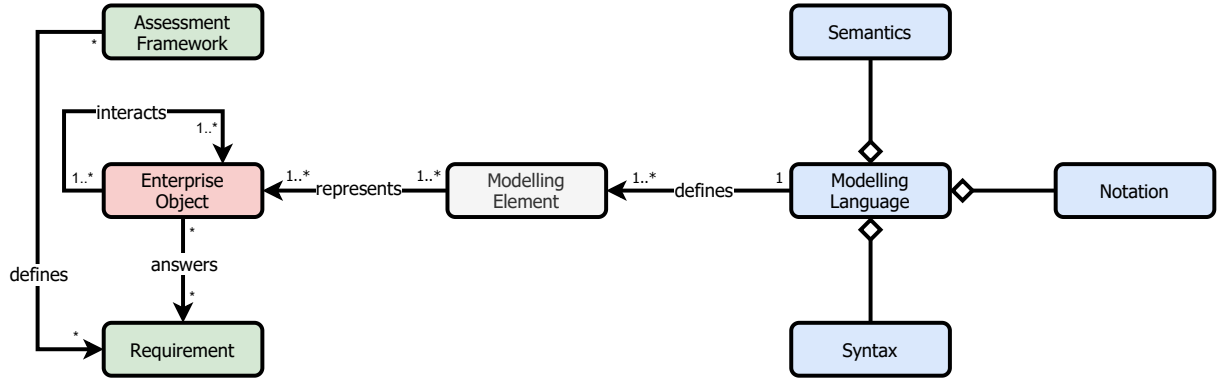


Figure 5.1: Relationship between the Requirement of the assessment framework and the Element of the modelling language.

Note that the meta-model is composed of the main elements presented in Figures 1.1 and A.1. The main element of the meta-model is the Enterprise Object (EO) and the interactions between EOs. In the context of this approach, an EO represents any tangible or intangible entity that can be represented through a modelling language element, so as to reflect the accomplishment of a requirement defined by an assessment framework. The naming EO has been previously introduced in [285] in the context of software engineering. Note that the interaction between EOs could also provide answer to the requirements. EOs and their interactions can be viewed as drivers of the achievement of requirements. The main intuition behind the proposed approach is that if a specific EO (or the interaction between EOs) can be represented through one or more elements of a modelling language, then the requirement can be answered by models defined using the elements of that language.

The relevance of EOs lies on the possibility to define concepts and their properties without the necessity to depend on a specific modelling language. Moreover, the measure of the quality of an EO depends on the level of abstraction used to represent it. In this sense, the level of abstraction of an EO must be such that the EO represents a concrete example of an enterprise element that can be further instantiated to modelling elements or their interactions (for EO interactions) from different modelling languages. This is essential to define if a language supports a certain requirement defined by some assessment framework.

An example of EO could be as follows: physical or digital work product. Note that the EO is abstract enough to be able to be represented using elements from different modelling languages. Considering, for instance, the ArchiMate language, the Work package, Deliverable, Business object, Data object, and Artefact elements can represent such EO. On the other hand, if BPMN is considered, the Data object element allows to represent the EO.

The evaluation process initiates with the selection of the modelling language M and the assessment framework F , which contains a list of requirements. Once F is chosen, the requirements defined by the framework must be decomposed into a list R of atomic requirements $[r_1, r_2, \dots, r_n] \in R$, which are defined as requirements that cannot be broken down into smaller ones, similar to atomic statements in logic [286].

After, a requirement $r \in R$ is selected and at least one Enterprise Object e must be generated with the objective to be linked to r . The condition of e is that it must be able to answer r . This allows to use EOs as connections between requirements and the modelling language that is being analysed. If there are one or more elements or interaction of elements of the language that can represent the EO, we assume that the requirement can be answered (at least partially) by the

language. If at least one e that answers r exists, the requirement is added to a list of supported requirements. Else, the requirement is considered as not supported by the language. This process must be repeated until all requirements in R have been analysed. Figure 5.2 presents this process graphically.

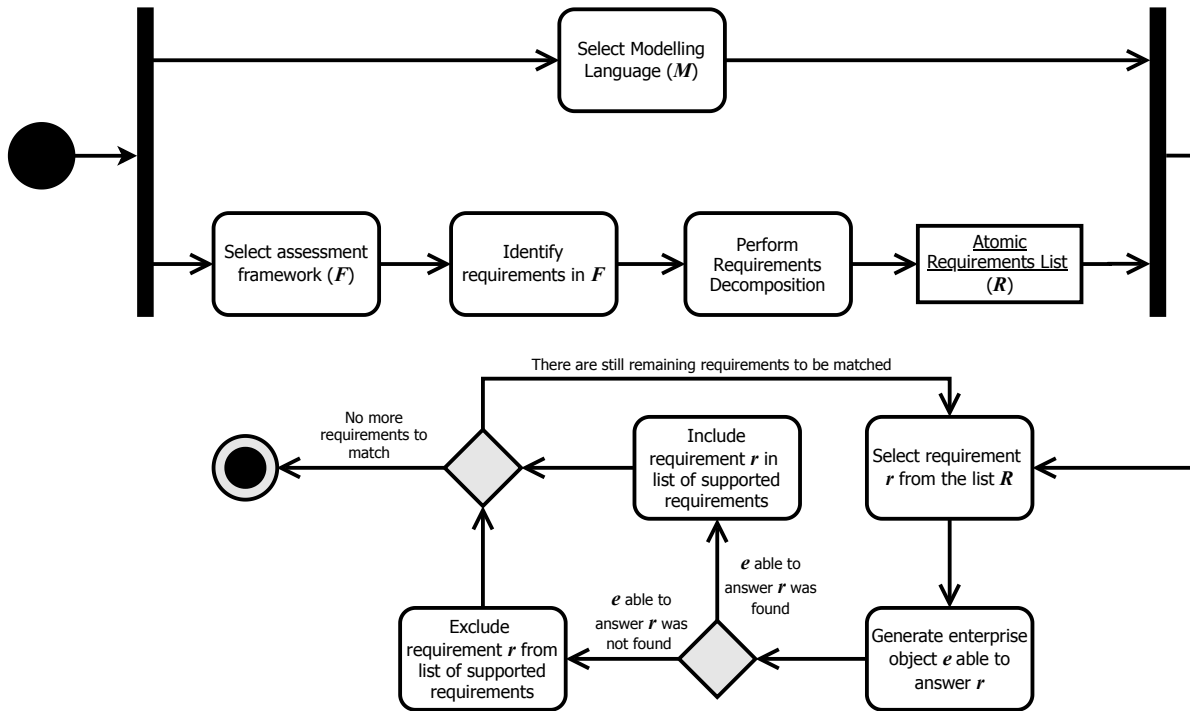


Figure 5.2: Overview of the evaluation process.

Among the steps of the proposed methodology, there are two particularly relevant tasks to provide satisfactory matching between requirements and modelling elements: (1) Requirements Decomposition, and (2) Requirement and modelling language matching. Those tasks are described in detail in the following sections.

5.3.1 Requirements Decomposition

This task is based on decomposing or refining a requirement with a high level of abstraction into indivisible requirements, named atomic requirements. These atomic requirements are represented using a formal mathematical expression. Formalised requirements allow them to be processed automatically or semi-automatically. In the scope of the proposed approach, they can be matched with EOs that are able to answer them.

The Requirements Decomposition method is based on the approach proposed in [278], which was successfully applied to different contexts in the literature such as software analysis [287] and interoperability assessment [286]. The method allows to perform requirements decomposition and formalisation through a data structure called Pseudo-Requirement Graph, which considers two types of objects: Pseudo-Requirements and Refinements. A pseudo-requirement is either a full requirement, a sub-requirement or an atomic requirement. The Refinement, on the other hand, holds the logical base information that is useful for the traceability of the decomposition process. The process is top-down, starting from the high-level requirements and ending with the formalisable atomic pseudo-requirements. Figure 5.3 presents an overview of the main tasks of

the process.

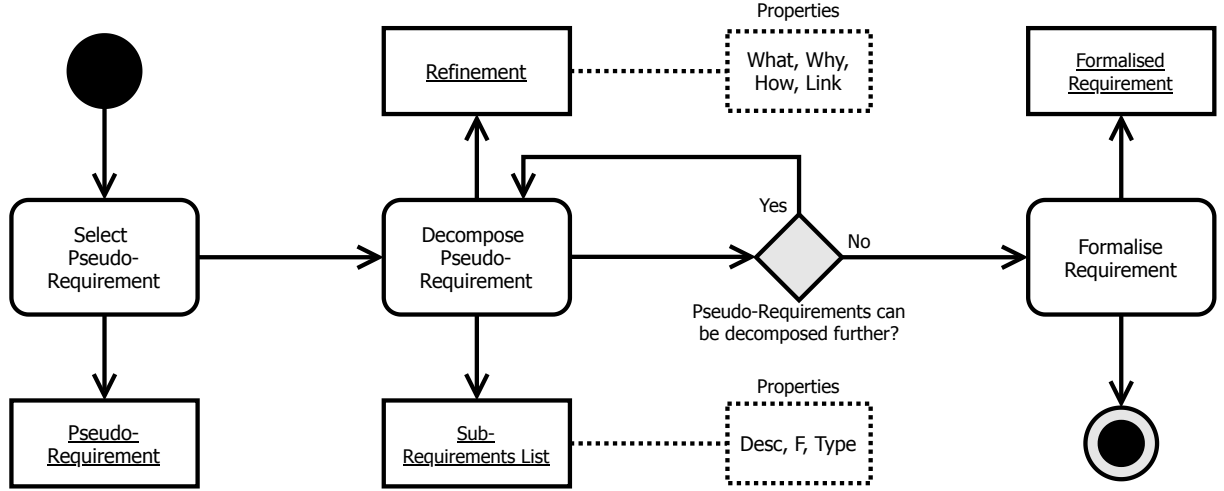


Figure 5.3: Overview of the requirements decomposition and formalisation through the Pseudo-Requirement Graph-based approach. Dotted lines and boxes are used to describe the properties of the output artefacts of the activities of the model.

The process initiates with the selection of the pseudo-requirement to be analysed, which is then decomposed into sub-requirements. Note that the elements of this approach (pseudo-requirements and refinements) are formally represented through the Pseudo-Requirement Graph. Each pseudo-requirement is associated to three items: **Desc**, its informal description; **F**, its formalisation in terms of CTL* [288], an extension of CTL (Computation Tree Logic) [289]; and finally its **Type**, that could be a **Req**, which is the proper requirement with a high abstraction level; a **Part**, which is a pseudo-requirement that is only a part of a requirement; and an **Atom**, which is a pseudo-requirement that cannot be decomposed any further. On the other hand, a refinement has four elements or properties: **What**, describing what part of the pseudo-requirement is refined; **Why**, explaining why the refinement is relevant; **How**, describing the result of modifying what is defined by the **What** property taking into consideration what is explained by the **Why** property; finally, the **Link** property is a CTL* formula describing how the initial Pseudo-Requirement is formally related to the refined one.

It is worth mentioning that there are categories of refinements; the most common ones are: Decomposition, which refers to a pseudo-requirement that is decomposed into more than one part; Precision, when the pseudo-requirement must be precised or clarified; Abstraction, when the pseudo-requirement is described in too much detail that it falls outside the scope of the system that is being studied; and Correction, which is applied when the pseudo-requirement is incorrect. Note that more categories can be defined depending on the needs of the problem to be addressed. On the other hand, when defining the CTL* formulas to formalise the requirements, logical connectors such as conjunction (\wedge), disjunction (\vee), implication (\rightarrow), or negation (\neg) can be used. Among the available CTL* connectors, two particularly relevant ones are the temporal modal operators **X**, which can be read as "next", and **U**, which can be read as "until". The former implies that the proposition ϕ has to hold at the next state if **X** is applied to it ($\mathbf{X} \phi$). The later means that for two propositions, ϕ and ψ , that are connected through **U** ($\phi \mathbf{U} \psi$), ϕ has to hold at least until some position ψ holds. This adds the possibility to formalise temporal conditions, which are often present in requirements definitions.

Figure 5.4 shows an example of the decomposition of a pseudo-requirement that starts as a full

requirement Rq . A refinement is applied and it has two categories: Precision and Abstraction, which could have been separated into two different refinements. This is not done in the example to maintain the simplicity. The refinement is of category Precision because a new requirement is derived from it ($Rq2$), and Abstraction because the original requirement is simplified to a new one with the same meaning ($Rq1$). These two pseudo-requirements are joint through the connector \mathbf{U} ($Rq2 \mathbf{U} Rq1$) since in order to identify competencies and roles for a standard process ($Rq1$), that standard process must be first defined ($Rq2$). Note that \mathbf{U} is specifically useful for this case since $Rq2$ must hold true until $Rq1$ takes place. The pseudo-requirement $Rq1$ is then decomposed to obtain $Rq11$ and $Rq12$, which are already atomic requirements. On the other hand, a refinement is applied on $Rq2$ of type Precision, since more requirements can be derived from it: the definition of a standard process implies the definition of its activities, sequence of activities, process outcomes, etc. Hence the requirement is extended to consider these aspects. Finally, a final refinement is applied to decompose $Rq2$ into six additional requirements that are atomic: $Rq21$ to $Rq26$.

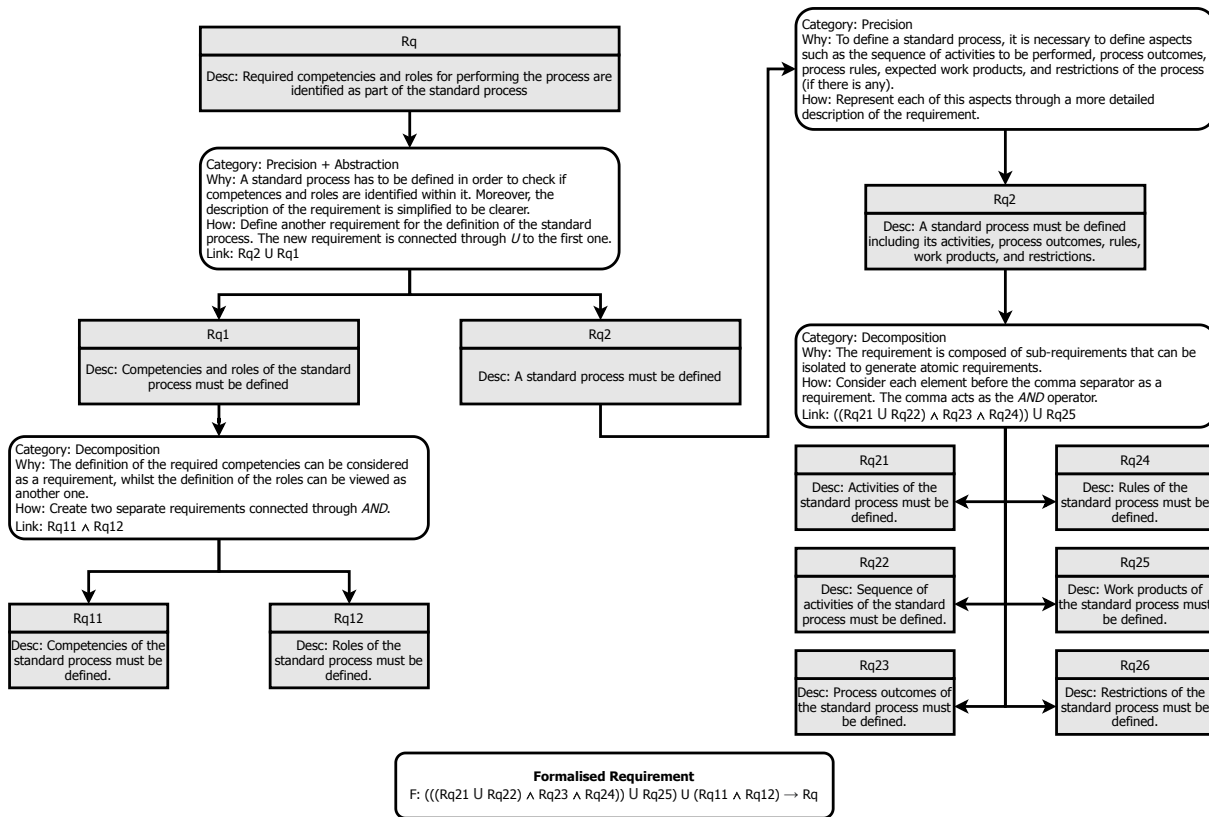


Figure 5.4: Example of the decomposition of a requirement Rq .

The result of the decomposition and formalisation process is a list of formalised atomic requirements that can be used in the next step of the methodology. The reader may refer to the original paper [278] for more details regarding the Pseudo-Requirement Graph approach. For simplicity, the atomic requirements obtained through the approach described in this section are named as requirements in the following sections.

5.3.2 Requirement and modelling elements matching

The core of the methodology is the requirement and modelling elements matching. It is based on the Goal Question Metric (GQM) paradigm [279], which is a systematic approach for defining and evaluating a set of goals through some measurement mechanism. The paradigm was used in [282] to assess the Generic Practices of the Capability Maturity Model Integration [18] that can be addressed through the use of Process Mining methods [283]. This approach is considered as a starting point of the requirement and modelling elements matching step of this work.

The application of GQM allows to have a model with three levels: Conceptual Level (Goal), in which a goal is defined for an object relative to some environment; Operational Level (Question), which defines a set of questions that are used to characterise the way the goal is achieved; and Quantitative Level (Metric), which defines a set of concrete data that is associated to the question and allows to answer it [279]. A Goal can derive one or more Questions, and the Question can derive one or more Metrics. Thus forming a hierarchical structure in which the level of abstraction is reduced from top to bottom.

Similarly to [282], considering the EO and requirement matching scenario, the **Goal** g represents whether a requirement r defined in the assessment framework F is achieved, and to which extent r is accomplished. The **Question** q is derived from each r . Ideally, each Requirement r should introduce only one question, since we part from the idea that requirements are already atomic. However, for certain requirements this may not hold true and they could be linked to more than one question. The **Metric** m is composed of one or more EOs e that allow to answer each q , if an e can be represented through elements from the modelling language M , then q is supported by M and g is visible through the language. Figure 5.5 graphically presents the relationships described before.

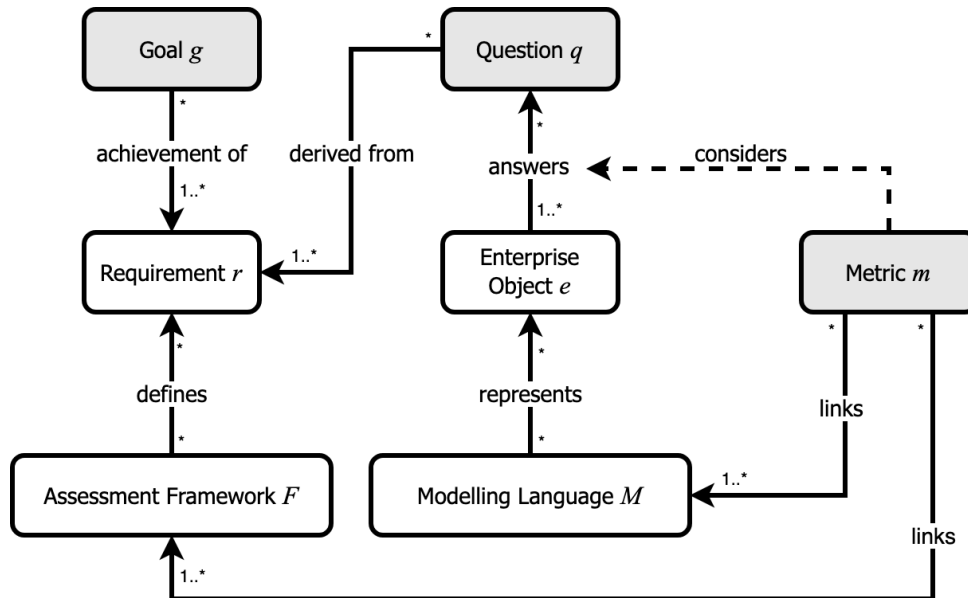


Figure 5.5: The Goal Question Metric paradigm within the scope of the proposed approach.

Note that once every visible g addressing a r is obtained, an aggregation method to define if some global requirement has been achieved can be applied, given that all r are atomic and, in some cases, they may have been part of a bigger requirement. However, aggregation is not considered in the proposed method since it depends on the specific F used and on the manner

that the results will be presented to the stakeholders.

5.4 Analysing BPMN and ArchiMate Considering ISO/IEC 33020

To validate the proposed approach, this section presents an analysis of the adequacy of the BPMN and ArchiMate modelling languages to answer requirements from the ISO/IEC 33020 standard Measurement Framework [16] for process capability assessment, described in Section 5.4.2. Some assumptions regarding the modelling languages used for the validation are introduced in Section 5.4.1. The expected result of the validation is a list of Process Attributes from the standard that can be answered through ArchiMate and BPMN models. The list is presented in Section 5.4.3 with an analysis of the results.

5.4.1 Considerations

Following the proposed approach, five **Assumptions** regarding BPMN are made for the validation:

- The elements of BPMN, version 2.0, without any extension are considered.
- BPMN models describing the Business Process are semantically and syntactically correct.
- There is at least one BPMN model describing the assessed Business Process that is the object of the assessment.
- The BPMN models are a reflection of the reality. Hence, information presented in the models are valid to answer the requirements defined by the assessment framework.
- The BPMN models are used to represent the assessed business process without considering any other external process that is not explicitly related to the assessed process.

Note that the last assumption is perhaps the most complex and it requires a more detailed explanation. This assumption is made considering that it is unlikely to find a BPMN model in an enterprise describing non-functional aspects of a process. In general, BPMN is used to describe the elements that are directly associated to the process execution. For instance, information related to goals or business strategies are not reflected, even though they could be somehow implicitly addressed using the modelling language. For instance, it is possible to define a BPMN model describing the sequence of tasks that are performed during the definition of the goals of the enterprise, thus reflecting a non-functional aspect of the assessed process, which may have a direct influence on the final process design and execution. Notwithstanding, the nonexistence of such models is assumed, as well as only the existence of models that explicitly reflect the execution aspects of the assessed process.

On the other hand, considering ArchiMate, the following **Assumptions** are made:

- The elements of ArchiMate, specification 3.1, without any extension are considered.
- The ArchiMate model describing the business process is semantically and syntactically correct.
- There is at least one model describing some aspect of the Business Process that is the object of the assessment. Elements from all aspects that can be used to represent a process are considered.

- The ArchiMate model is a reflection of the reality. Hence, it presents the As-Is state of the process, not the To-Be.

5.4.2 Process Attributes and Indicators of ISO/IEC 33020

ISO/IEC 33020 is an International Standard that defines a Measurement Framework for measuring the process quality characteristic of process capability [16]. It defines Process Capability on a six point ordinal scale with a hierarchy based on the following levels: Level 0 (Incomplete Process), Level 1 (Performed Process), Level 2 (Managed Process), Level 3 (Established Process), Level 4 (Predictable Process), and Level 5 (Innovating Process). Each level defines a set of Process Attributes (PAs) and the framework establishes a set of Indicators that represent their full achievement. For instance, for Level 2 (Managed Process), one of the PAs is the Work Product Management Process Attribute, which is a measure of the extent to which the work products produced by the process are appropriately managed, and it contains the following Indicators of the achievement: (1) requirements for the work products of the process are defined; (2) requirements for documentation and control of the work products are defined; (3) work products are appropriately identified, documented, and controlled; (4) work products are reviewed in accordance with planned arrangements and adjusted as necessary to meet requirements. In the context of the methodology introduced in this chapter, the Indicators of the framework are analogous to the requirements that must be refined during the first step of the approach.

Table 5.1 presents the PAs from the framework and the number of Indicators for each of them, a total of 43 Indicators are defined in the standard. Note that Level 0 is excluded in the table, since at that level the standard defines that there is little or no evidence of any systematic achievement of the process purpose. Hence, no evidence of achievement of PAs are present to be analysed.

Table 5.1: Number of Indicators per Process Attribute (PA) in the ISO/IEC 33020 Measurement Framework. The first number of the Id in each row corresponds to the Capability Level in which the PA is grouped.

Id	Process Attribute	Number of Indicators
1.1	Process performance	1
2.1	Performance management	8
2.2	Work product management	4
3.1	Process definition	5
3.2	Process deployment	6
4.1	Quantitative analysis	7
4.2	Quantitative control	5
5.1	Process innovation	4
5.2	Process innovation implementation	3
Total		43

The ISO/IEC 33020 framework also provides a rating scale and a rating method for the PAs, as well as an aggregation method to define the process capability level. However, these aspects are not within the scope of this work and they are not addressed in this chapter, since the proposed approach is specifically focused on the treatment of requirements (PA Indicators in the validation). On the other hand, although the standard defines Indicators up to Level 5,

only those from Levels 1 and 2 are used to validate the proposed approach, this is considered as sufficient to demonstrate the functioning of the method.

5.4.3 Results and Discussion

The methodology proposed in Section 5.3 was applied for each Process Attribute defined for the Levels 1 and 2 of the ISO/IEC 33020 Measurement Framework. The objective was to assess which Indicators of the PAs can be answered through BPMN and ArchiMate models.

In the context of the method validation, the Indicators of the standard are considered as the original requirements. The first step was based on decomposing those Indicators to obtain atomic requirements. An example of requirement decomposition is shown in Table 5.2, which presents the decomposed requirements from the Indicator 2.1.a of PA 2.1, it is based on the requirement decomposition method presented in Section 5.3.1. The table presents the **What**, which is the pseudo-requirement to be decomposed. Note that in the first round of the decomposition process, the pseudo-requirement is the Indicator. The **Type** of decomposition is Precision in the first round and Decomposition in the second round. The **Why** explains the relevance of applying the decomposition on the requirement. The **How** explains details regarding how the decomposition is performed. In round one, the original requirement is extended to consider performance in quantitative and qualitative terms. In round two, the extended requirement is decomposed so as to reach the atomic state. The **Link** formally defines the new requirement. Note that in the second round, the link shows a connection between the two atomic requirements *Rq21a1* and *Rq21a2* through the logical \wedge connector. This is particularly relevant when analysing the global achievement of the original requirement: with the \wedge , both atomic requirements must be fulfilled to satisfy the original requirement *Rq21a*. If other connectors are used, the conditions for satisfying it could vary.

Once the list *R* composed of atomic requirements related to the Indicators of the standard is obtained, the capability of the modelling language *M* to support the requirements was analysed. For each atomic requirement *r*, a Goal *g* is defined, which in this case always refers to the capability of the language to represent an EO related to *r*. After, the Question *q* is defined, which is derived from *g* and it introduces the expected EO that is linked to *g*. Note that, in some cases, *g* already introduces the EO and *q* is only a re-definition of *g* using question marks. Specially when *g* is simple enough. However, this is not always the case and *q* allows to define the EOs and EO interactions required to achieve *g*. Finally, the Metric *m* allows to define or specialise concrete EOs and EO Interactions that must be able to be represented by the language. If they cannot be represented through the modelling elements, then *r* is not supported by *M*. Tables 5.3 and 5.4 show the results for some requirements considering the ArchiMate language. Table 5.3 presents a scenario in which the requirement is able to be answered by the language, whilst 5.4 shows the opposite situation.

Note that, in some cases, more than one atomic requirement can be extracted from an Indicator. For instance, the number of Indicators of PA 2.2 is equal to four, from which a total of 14 atomic requirements were extracted. Table 5.5 shows the requirements extracted from each Indicator of the PA. It is worth mentioning that some atomic requirements that are equivalent could also be obtained, such is the case for *Rq22a1*, *Rq22b1*, *Rq22c1*, and *Rq22d1*; and also for *Rq22a2* and *Rq22d2*.

The table also introduces the formal representation of the Indicator, using as source the atomic requirements decomposed from it. For instance, for Indicator *Rq22b*, the expected work products must be first defined (*Rq22b1*), and it must hold until the requirements for documentation and control of the expected work products are defined (*Rq22b2* and *Rq22b2*). This is

Requirement	Type	Why	How	Link	Requirements	Round
Rq21a. Objectives for the performance of the process are identified.	Precision	The performance could be qualitative or quantitative. The requirement must be defined accordingly.	Include description of qualitative and quantitative performance in the requirement.	<i>Rq21a</i>	Rq21a. Qualitative and quantitative objectives for the performance of the process are identified.	1
Rq21a. Qualitative and quantitative objectives for the performance of the process are identified.	Decomp	There are two requirements, one referring to qualitative performance objectives and the other to quantitative ones.	Two new requirements are derived from Rq21a.	<i>Rq21a1</i> \wedge <i>Rq21a2</i>	Rq21a1. Qualitative objectives for the performance of the process are identified. Rq21a2. Quantitative objectives for the performance of the process are identified.	2

Table 5.2: Requirements decomposed from the first Indicator of Process Attribute 2.1 (Performance management) from the ISO/IEC 33020 standard. A precision of the Indicator has been made in round one of the decomposition, and a decomposition has been performed in round two, generating two new requirements from the initial Indicator. The full formalisation of the Indicator is as follows: $Rq21a1 \wedge Rq21a2 \Rightarrow Rq21a$.

logically represented using the connectors \cap and \wedge in the formalisation. Note that this representation may not be relevant when analysing modelling languages, since the idea is that all atomic requirements must be able to be represented to check the capability of the language to answer the Indicator (global requirement). However, they gain utility when performing actual assessments, not only using enterprise models but any form of assessment evidence. For the example described above, if *Rq22b1* does not hold true, there would be no necessity to check *Rq22b2* and *Rq22b3*: if the expected work products are not defined, aspects such as documentation and control requirements for those work products cannot be defined either.

Since the objective was to answer which Indicators from each PA of the standard can be answered through a BPMN or ArchiMate model, an aggregation method was required to obtain a final result per Indicator considering the atomic requirements extracted from it. Once the result (defining if the requirement can be answered through some modelling element) was obtained for all requirements r in a list R originated from the Indicator, they were aggregated considering the aggregation function s presented in Equation 5.1.

Requirement r	Rq21d11. Qualitative performance of the process must be adjusted to meet qualitative performance plans.
Goal g	Activities to adjust qualitative performance and their link to plans can be represented.
Question q	Can adjustment activities of qualitative performance and their link to plans be represented?
Metric m	EO defines an activity to adjust the process considering the EOs defining qualitative performance plans. EOI defines interactions between those adjustment activities. They must be linked to EOs defining performance objectives or plans.
Modelling Element	Business function, Business interaction, Course of action, Contract, Assignment relationship.
Supported Language	by Yes

Table 5.3: Results of the analysis of the ArchiMate language for the requirement Rq21d11, decomposed from the Indicator 2.1.a of PA 2.1 of the ISO/IEC 33020 standard.

$$s = \begin{cases} \text{Yes,} & \forall r : (r \text{ has } e) \wedge (e \text{ supported by } M) \\ \text{Partially,} & \exists r : (r \text{ has } e) \wedge (e \text{ supported by } M) \\ \text{No,} & \text{otherwise} \end{cases} \quad (5.1)$$

The Indicator is supported by the modelling language M if all requirements have EOs e that are supported by M ($s = Yes$) through its modelling elements. If at least one r is supported (but not all of them), then the Indicator is Partially visible by M ($s = Partially$). Finally, if there is no EO supported by the M , the Indicator is not visible ($s = No$). Figure 5.6 presents the overall results for the PAs of the standard from Levels 1 and 2, considering BPMN and ArchiMate.

Results for **BPMN** show that the first PA of the ISO/IEC 33020 standard is supported by the language. Within the Capability Level 1 (Performed process), the PA 1.1 (Process performance) contains only one Indicator, from which two atomic requirements were extracted. Note that this Indicator is specifically focused on the outcomes of a business process and its execution, which are aspects that can be naturally viewed through BPMN models, since they are specifically devised to represent business process execution aspects. For Capability Level 2 (Managed process), the PA 2.1 (Performance management) contains 8 indicators, from which 32 atomic requirements were decomposed. Results show that two Indicators are supported by BPMN, four are partially supported and two are not supported. Note that this PA refers to aspects regarding performance and work product management. In this sense, the Indicators that are supported by BPMN are those regarding monitoring activities during process execution, the existence of plans and monitoring activities related to process performance. These aspects can be addressed through BPMN elements such as Pools, Participants, Activities, Data objects, Data stores, among others. However, more non-functional aspects such as the extent to which the participants of the process are prepared to execute their responsibilities cannot be represented. On the other hand, the PA 2.2 (Work product management) puts emphasis on managing work products through the requirements definition, control activities, identification of relevant work products, among other aspects. A total of 14 atomic requirements were extracted from the four Indicators of this PA. Results show that one Indicator, the one referring to the identification, documentation and

Requirement r	Rq21f1. Personnel performing the process must have formation for executing their responsibilities.
Goal g	The formation of personnel in relation to their responsibilities can be represented.
Question q	Can objects describing the formation of individuals, with relation to their responsibilities, be represented?
Metric m	EO representing a set of formations of an individual, or an individual formation. EOI relating a formation or set to an EO defining individual responsibility or a set of responsibilities.
Modelling Element	Not possible to represent formation EO or EO Interaction representing formation of individuals.
Supported Language	by No

Table 5.4: Results of the analysis of the ArchiMate language for the requirement Rq21f1, decomposed from the Indicator 2.1.a of PA 2.1 of the ISO/IEC 33020 standard.

control of work products can be answered through an analysis of a model, considering that these activities can be represented in BPMN in execution time through Tasks, Data Objects, Data Stores, etc. The rest of the Indicators can only be partially answered through BPMN models. On the other hand, considering the **ArchiMate** language, results show that, similarly to BPMN, the elements of ArchiMate are able to represent EOs linked to requirements from PA 1.1 satisfactorily. However, it allows to answer more Indicators from PA 2.1 (with six Indicators that can be fully viewed through ArchiMate models and only one that can be Partially viewed). Moreover, it gives full support to view the accomplishment of the atomic requirements derived from the Indicators of PA 2.2.

The method validation presented in this section allowed to validate the proposed approach and it provides a concrete view of the capabilities of BPMN and ArchiMate to answer Indicators defined in the ISO/IEC 33020 Measurement Framework for Process Capability Assessment from Levels 1 and 2. Results showed that models defined using the ArchiMate language provide an extended view of the requirements derived from the Indicators of the standard when compared to BPMN. Notwithstanding, since not all Indicators can be answered through the models defined in ArchiMate, other sources of information must be analysed in order to address them when performing assessments. Moreover, note that the conclusions regarding the studied modelling languages are *a priori*, which means that the language provides elements that can be used to describe the requirements but it does not imply that those requirements are actually present in the models defined by an enterprise. In such case, a detailed analysis of the concrete models must be performed, which can also be done by following the proposed methodology. This aspect is addressed in the following section.

5.5 Assessing Capability Level of a Business Process

The method validation presented in Section 5.4 showed the Indicators extracted from the ISO/IEC 33020 standard, considering Capability Levels 1 and 2, that can be answered through BPMN and ArchiMate models. In this section, a case study based on assessing two models (one for each language) describing a specific process so as to define which Indicators from the standard are met

Indicator	Requirement	Formalised
Rq22a. Requirements for the work products of the process are defined.	Rq22a1. Expected work products of the process must be defined. Rq22a2. Requirements for the work products of the process must be defined.	$Rq22a1 \cup Rq22a2 \Rightarrow Rq22a$
Rq22b. Requirements for documentation and control of the work products are defined.	Rq22b1. Expected work products of the process must be defined. Rq22b2. Requirements for documentation of the work products must be defined. Rq22b3. Requirements for control of the work products must be defined.	$Rq22b1 \cup (Rq22b2 \wedge Rq22b3) \Rightarrow Rq22b$
Rq22c. Work products are appropriately identified, documented, and controlled.	Rq22c1. Expected work products of the process must be defined. Rq22c2. Work products are appropriately identified. Rq22c3. Work products are appropriately documented. Rq22c4. Work products are appropriately controlled.	$Rq22c1 \cup (Rq22c2 \cup (Rq22c3 \wedge Rq22c4)) \Rightarrow Rq22c$
Rq22d. Work products are reviewed in accordance with planned arrangements and adjusted as necessary to meet requirements.	Rq22d1. Expected work products of the process must be defined. Rq22d2. Requirements for the work products of the process must be defined. Rq22d3. Arrangements for work products of the process must be planned. Rq22d4. Work products must be adjusted to meet requirements. Rq22d5. Work products must be reviewed in accordance with planned arrangements.	$Rq22d1 \cup ((Rq22d2 \cup Rq22d4) \wedge (Rq22d3 \cup Rq22d5)) \Rightarrow Rq22d$

Table 5.5: Requirements extracted from each Indicator of Process Attribute 2.1 (PA 2.1 Performance management) from the ISO/IEC 33020 standard.

by it is described. Specifically, the assessment of one capability level for the process is performed. During the analysis, for each Indicator a rating is given based on the scale from ISO/IEC 33020, which is defined as follows: N - Not Achieved, P - Partially Achieved, L - Largely Achieved, F - Fully Achieved, as it was explained in Chapter 1.

The process to be evaluated is regarding Samples Management (SM) in a public organisation. It comprises the activities of reception, treatment, destruction, and returning of different types of chemical samples that are used in the institution. This is the same scenario that was considered for the case study of Chapter 4. In this section, the content of each model is analysed in order to assess if the process complies with the Indicators that are object of analysis. For this case study, the following assumptions are made:

- The model is a reflection of the assessed process. Hence, it is defined with fidelity to the reality showing the As-Is state, not the To-Be.
- The requirements from the assessment framework that are not able to be seen through the models must be analysed through other means by the assessors. This is not performed in

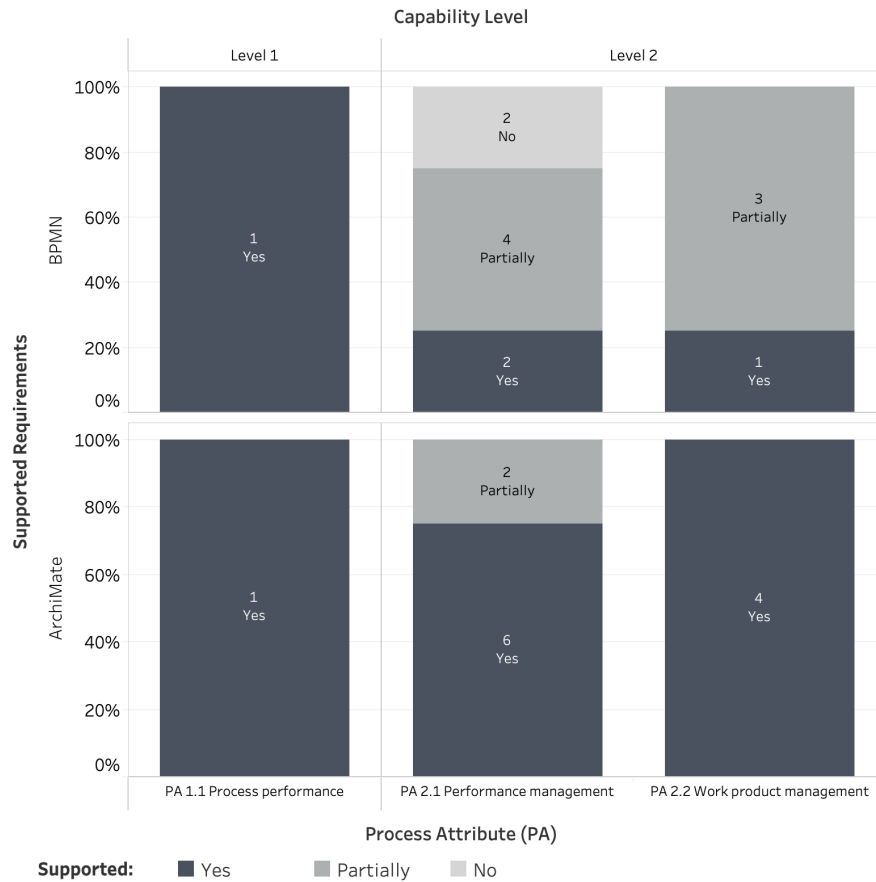


Figure 5.6: Results obtained through the analysis of the BPMN and ArchiMate modelling languages following the proposed methodology.

this case study, since it is not within the scope of this work. In the case that there is not enough information to provide a proper rating for an Indicator, the N - Not Achieved rating will be given, considering that the model is the only source of evidence for the assessment.

- Although the activities and tasks from the process are the same for both models. The ArchiMate model has more elements from different layers besides the Business Layer in order to take advantage of all available modelling elements from the language.

The rest of this section presents the analysis of the BPMN model. After, the results of the analysis of the ArchiMate model are described. Finally, a brief discussion addressing relevant aspects of the case study is presented.

5.5.1 BPMN

The BPMN model is presented in Figure 5.7. The process starts from the definition of the samples that are needed. After, the validation and request of the samples is performed. Then the samples are received, checked, and deposited in a cold room. The samples are then used when necessary and destroyed when it is no longer required. Note that there are sub-processes such as the use of the samples and the destruction, which are not analysed since they are out of the scope of the objective of the assessed process.

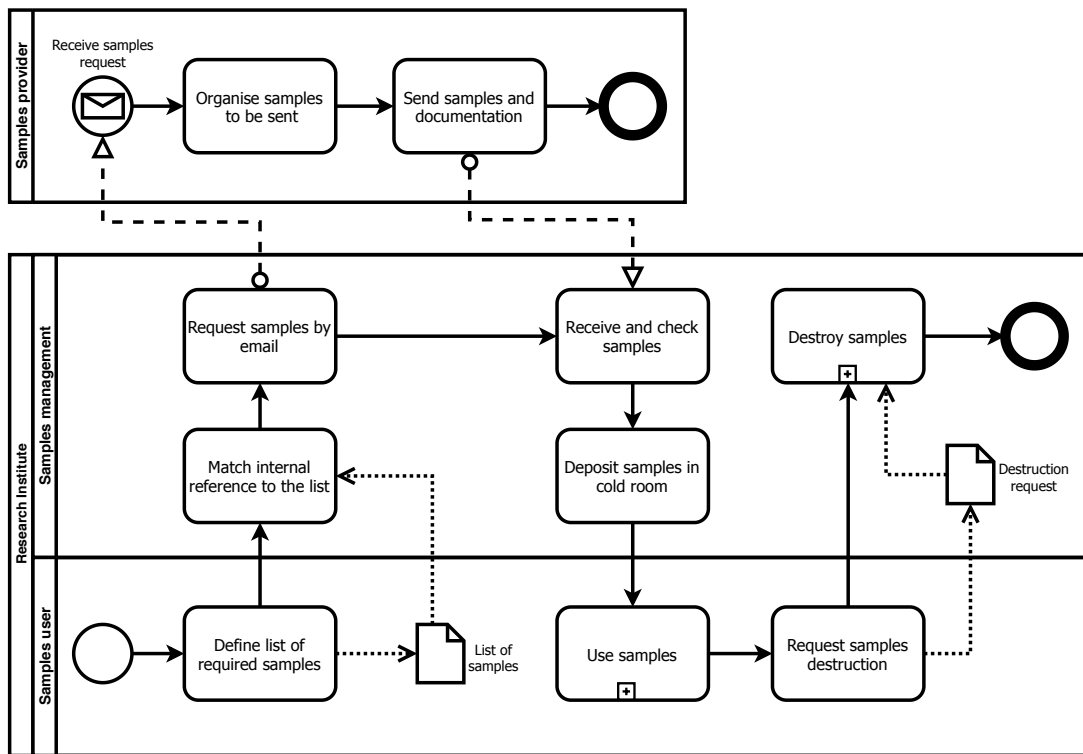


Figure 5.7: BPMN model describing the Samples Management Process.

Since it is already known which Indicators can be viewed through BPMN models, the focus is put only on them for the analysis. Moreover, for simplicity, only the Indicators from Capability Level 2 are considered, namely: PA 2.1.b. Performance of the process is planned, PA 2.1.c - Performance of the process is monitored, and PA 2.2.c - Work products are appropriately identified, documented, and controlled. These Indicators can be fully viewed through BPMN models according to the results obtained in Section 5.4.

To analyse each Indicator, the available information obtained during the analysis of the modelling languages is used, which was presented in Section 5.4. Specifically, the elements that are used are the Metric, which is defined by the expected EO for a requirement derived from the Indicator, and the Modelling Element, which is a concrete expected modelling element from the modelling language to represent the EO or the EO interaction. The main task when performing the assessment is to search the elements in the model, analyse those that may refer to the requirement, and provide the rating for the requirement according to their perception. Table 5.6 presents an example of the information that was used to give the ratings for the Indicators derived from PA 2.1.

It is worth mentioning that when the atomic requirements from PA 2.1 are analysed, most of them refer to qualitative or quantitative performance. In this sense, it is necessary to define or further specialise both concepts so as to place them in the context of the specific process that is the object of the analysis. Hence, the following definitions are presented:

- **Quantitative Performance for SM Process:** two main quantitative performance aspects are considered, *time*, which aims at enhancing the time to perform the tasks and communicate between parties; and *cost*, which aims at optimising the costs of the operations related to the process. For SM, the time for the definition of the list of required

5.5. Assessing Capability Level of a Business Process

Atomic Requirement (r)	Re-	Metric (m)	Element	In Model	Element Details	Rating
Rq21b1.	Qualitative performance plans must be defined.	EO contains guidelines to reach the not measurable (qualitative) objectives of the process.	Data object, data store.	No	-	N
Rq21b2.	Quantitative performance plans must be defined.	EO contains guidelines to reach the measurable (quantitative) objectives of the process.	Data object, data store.	No	-	N
Rq21c11.	Qualitative performance monitoring activities must be defined.	EO defines a monitoring activity to check qualitative performance of the process. EOI defines interactions between monitoring activities to check qualitative performance. They could be linked to EOs defining performance objectives or plans.	Activity, Task, Event.	Yes	Receive and check samples (Task). Matching of the internal reference list of samples (Task).	P
Rq21c12.	Quantitative performance monitoring activities must be defined.	EO defines a monitoring activity to check quantitative performance of the process. EOI defines interactions between monitoring activities to check quantitative performance. They must be linked to EOs defining performance objectives or plans.	Activity, Task, Event.	No	-	N
Rq21c21.	Qualitative performance monitoring activities must be performed.	EO defines a monitoring activity to check qualitative performance of the process. EOI defines interactions between monitoring activities to check qualitative performance. They could be linked to EOs defining performance objectives or plans.	Activity, Task, Event.	Yes	Receive and check samples (Task). Matching of the internal reference list of samples (Task).	P
Rq21c22.	Quantitative performance monitoring activities must be performed.	EO defines a monitoring activity to check quantitative performance of the process. EOI defines interactions between monitoring activities to check quantitative performance. They must be linked to EOs defining performance objectives or plans.	Activity, Task, Event.	No	-	N

Table 5.6: Ratings given to the requirements decomposed from the Indicators 2.1.b and 2.1.c from Process Attribute 2.1 of the ISO/IEC 33020 standard. The table also shows the metric, the expected elements to answer the requirement, the presence of a concrete instance of the element in the model, and the details of that instance in the case it exists.

samples must be reduced, the preparation and organisation of samples to be sent must be optimal, the materials and methods used for samples destruction must cost only the necessary, etc.

- **Qualitative Performance for SM Process:** regarding performance from a qualitative view, regulations and guidelines for samples handling and destruction must be followed, internal references of samples must be used, the quality of the received samples must be checked, etc.

Results for Indicator PA 2.1.b show the absence of elements representing quantitative performance plans. Therefore, the rating given to the requirements decomposed from this Indicator is N - Not Achieved (*Rq21b1* and *Rq21b2*). On the other hand, regarding Indicator PA 2.1.c, there is a presence of tasks for matching the internal reference list of samples and for checking the samples once received by the suppliers, which implies the existence of some monitoring capability in terms of process quality. However, there is a lack of sufficient monitoring activities for other aspects of the process, and the description of the existing ones is rather superficial. A rating of P - Partially Achieved is given to the atomic requirements *Rq21c11* and *Rq21c21*, under the assumption that the monitoring activities are performed repeatedly. On the other hand, there is no evidence of the existence of quantitative performance monitoring activities, thus the rating given to the requirements *Rq21c12* and *Rq21c22* is N - Not Achieved.

Finally, considering PA 2.2, specifically for the Indicator 2.2.c, four requirements were extracted: (1) Expected work products of the process must be defined (*Rq22c1*), (2) Work products are appropriately identified (*Rq22c2*), (3) Work products are appropriately documented (*Rq22c3*), and (4) Work products are appropriately controlled (*Rq22c4*). The results of the analysis for these requirements are presented in Table 5.7.

The same strategy that was applied for the Indicators of PA 2.1 was followed, giving one rating to each requirement. Note that the four requirements of Indicator 2.1.c are strongly focused on work products.

Therefore, the main idea was to consider the elements in the model that are EOs representing work products. In this sense, the possible modelling elements of BPMN to represent work product EOs are Data Objects, Data Stores, and specific work products described in a Task, as shown in the column *Modelling Element* in Table 5.7. Specifically, three elements that are related to work products were identified: the list of required samples, the destruction request, and the samples request email. The first two are Data Objects, whilst the later is a work product described in the detail of a Task. It is noticeable that the definition and identification of the work products happens at the moment the work product is included in the enterprise model. The reasoning behind this is that if the element is included in the model, it has already been identified by the individuals that are responsible of the process. Hence, for this particular case, the results for *Rq22c1* and *Rq22c2* are similar. Note that this would not apply if other assessment evidences are considered, the existence of separate requirements for the definition and identification of work products is therefore necessary. The same rating is set for *Rq22c1* and *Rq22c2*, which is equal to P - Partially Achieved. This rating was given considering that there exist elements in the model that act as output or input of process activities that act as work products, which were enumerated before. However, several other work products can be defined and identified in the context of the process, such as guidelines for the definition of required samples, samples request templates, a framework for the reception and quality checking of the received samples, among others. Hence, although there are work products that allow to execute the process, they are not considered as sufficient to reach a higher rating. On the other hand, considering requirement *Rq22c3*, there are two elements that show the presence of documentation of work products, presented in the

Atomic Requirement (r)	Requirement	Metric (m)	Modelling Element	In Model	Element Details	Rating
Rq22c1.	Expected work products of the process must be defined.	EO defining one or more physical or digital work product.	Work product described in Task. Data object, Data store.	Yes	List of required samples (Data object). Samples request email (Task description). Destruction request (Data object).	P
Rq22c2.	Work products are appropriately identified.	EO defining one or more physical or digital work product.	Work product described in Task. Data object, Data store.	Yes	List of required samples (Data object). Samples request email (Task description). Destruction request (Data object).	P
Rq22c3.	Work products are appropriately documented.	EO representing a physical or digital document addressing work products configuration.	Data object, Data store.	Yes	List of required samples (Data object). Destruction request (Data object).	P
Rq22c4.	Work products are appropriately controlled.	EO is an activity for controlling work product EOs. EOI is the interaction between the controller activity and the controlled work product in the process.	Data object, Data store (work product EO). Task, Event (for control activity EOs).	No	-	N

Table 5.7: Ratings given to the requirements decomposed from the Indicator 2.2.c from Process Attribute 2.2 of the ISO/IEC 33020 standard. The table also shows the metric, the expected elements to answer the requirement, the presence of a concrete instance of the element in the model, and the details of that instance in the case it exists.

third row of Table 5.7. Notwithstanding, since there is a lack of defined work products, this also implies a lack of documentation. Moreover, for the specific case of the email for requesting samples, the content of the email (or of any other communication mean) seems to be defined in a rather ad-hoc manner, without the use of any template or document with guidelines linked to the task and that specific work product. Considering these issues, a rating equal to P - Partially Achieved is given to the requirement. Finally, regarding requirement *Rq22c4*, a N - Not Achieved rating has been given considering that there is a lack of control activities for the work products

of the process.

Once the ratings to the atomic requirements have been given, they were aggregated to obtain the final rating for each Indicator. The aggregation method was based on averaging the ratings given to the requirements considering numerical values, defined as follows: F - 4, L - 3, P - 2, N - 1. The final results for the Indicators are N - Not Achieved for the Indicators PA 2.1.b and PA 2.1.c, and P - Partially Achieved for the Indicator PA 2.2.c.

5.5.2 ArchiMate

The model is presented in Figure 5.8. The process is similar to the one presented in Figure 5.7 with the main difference that the process has the representation of the roles. In BPMN, roles are represented using the lanes, in ArchiMate, there are specific elements that are associated to the business functions and the business actors. Besides this divergence, information about the process is more detailed since there are elements that are specific from ArchiMate such as Goals, Requirements, Deliverables, Work Packages, Application Services, and Application Components.

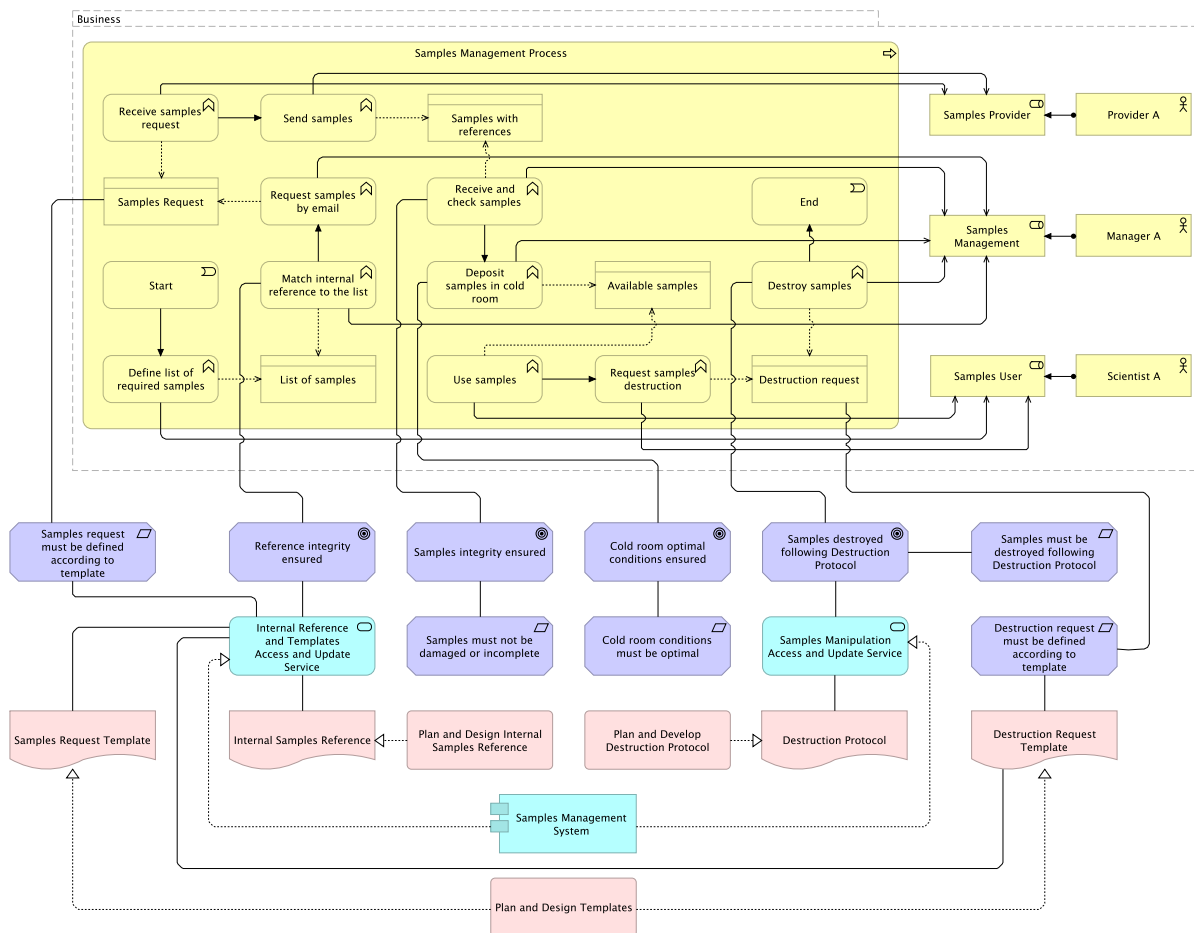


Figure 5.8: ArchiMate model describing the Samples Management Process.

Similarly to BPMN, as a result of the application of the method presented in Section 5.4, there is already available information regarding the Indicators of ISO/IEC 33020 that can be assessed through enterprise models defined using ArchiMate. Moreover, only the Indicators from Capability Level 2 are considered for the sake of simplicity.

Considering PA 2.1, two Indicators cannot be fully viewed through ArchiMate models: PA 2.1.a.f. Personnel performing the process are prepared for executing their responsibilities; and PA 2.1.a.g. Resources and information necessary for performing the process are identified, made available, allocated and used. These are excluded from the analysis in the case study. The rest of the Indicators of the PA were analysed, these are: PA 2.1.a. Objectives for the performance of the process are identified; PA 2.1.b. Performance of the process is planned; PA 2.1.c. Performance of the process is monitored; PA 2.1.d. Performance of the process is adjusted to meet plans; PA 2.1.e. Responsibilities and authorities for performing the process are defined, assigned and communicated; and PA 2.1.h. Interfaces between the involved parties are managed to ensure both effective communication and clear assignment of responsibility. As for PA 2.2, all Indicators were considered since they are analysable through ArchiMate models. The Indicators are: PA 2.2.a Requirements for the work products of the process are defined; PA 2.2.b. Requirements for documentation and control of the work products are defined; PA 2.2.c. Work products are appropriately identified, documented, and controlled; PA 2.2.d. Work products are reviewed in accordance with planned arrangements and adjusted as necessary to meet requirements. The ratings that were given for the Indicators of the PAs are presented in Figure 5.9. The figure also shows the number of Requirements from a certain Indicator that achieved a certain Rating. For instance, the Indicator PA 2.1.a. has one Requirement that has a Rating equal to N - Not Achieved, and another Requirement with Rating equal to L - Largely Achieved.

Results for PA 2.1 show several N - Not Achieved ratings for different atomic requirements. Specially for the requirements extracted from the Indicator 2.1.h, which makes reference to the management of interfaces between parties. All requirements of this Indicator obtained a rating of N - Not Achieved since no evidence of management of interfaces in the process was found. On the other hand, some atomic requirements were given the L - Largely Achieved rating. As an example, for requirement Rq21a1. Qualitative objectives for the performance of the process are identified from PA 2.1.a. The Metric is defined as follows: It is required an EO that represents an objective, goal, target or intention regarding the performance of the process defined in a not measurable manner. In this context, the following elements were found in the model: Reference integrity ensured (Goal element in the model); Samples integrity ensured (Goal); Cool room optimal conditions ensured (Goal); and Samples destroyed using protocol (Goal). These elements show that there are some objectives or goals regarding qualitative performance of the process. However, the use of samples could follow certain regulations or protocols, whilst assuring samples integrity could follow more concrete or specific quality requirements. These details do not allow a higher rating for achievement of this requirement. The same principle was followed by the assessors to define the same rating for the requirement Rq21b1. Qualitative performance plans must be defined. For this requirement, the Metric is defined as follows: EO contains guidelines to reach the not measurable (qualitative) objectives of the process. The assessors have identified the following elements in the model: Samples must be destroyed following destruction protocol (Requirement element); Destruction protocol (Deliverable element); Samples request template (Deliverable); and Requirement elements for samples request, samples integrity. Note that the evaluation considers both, the Requirement elements of ArchiMate and the Deliverables connected to those requirements, acting as plans or guidelines to achieve those requirements. In this sense, there are some missing plans such as a protocol to ensure optimal conditions of the cold room, or a planned evaluation approach for checking integrity of the samples. Considering the P - Partially Achieved rating, it was given to three requirements, two of them extracted from the Indicator PA 2.1.c, which refers to monitoring activities of the performance of the process. In this sense, the elements found in the model to support the achievement of the requirements of this Indicator are the Business Function elements linked to elements representing plans and objectives

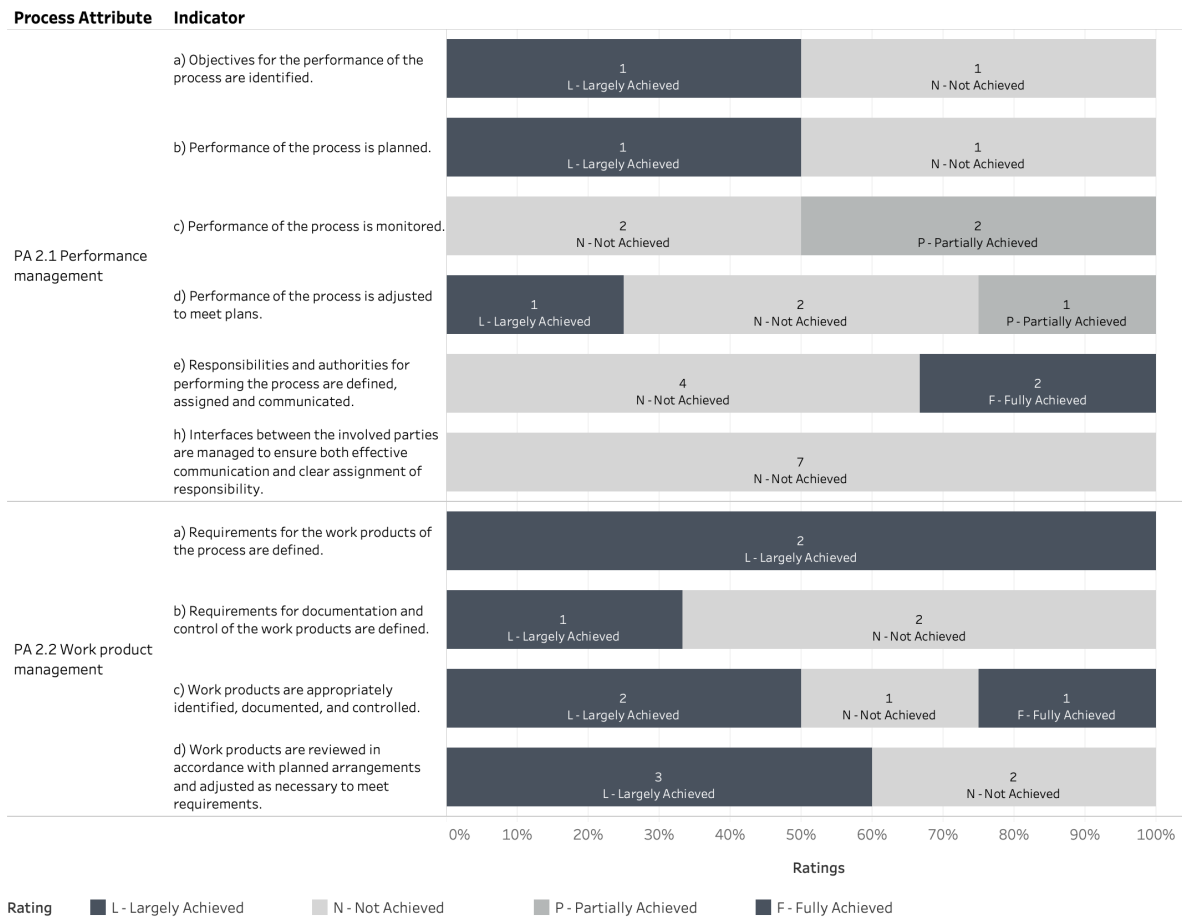


Figure 5.9: Ratings given to the Indicators of the Process Attributes 2.1 and 2.2 of the ISO/IEC 33020 for the Samples Management Process, considering the ArchiMate model.

EOs. The functions that have links to plans or objectives are: *Match internal reference to list*; *Receive and check samples*; and *Destroy samples*. These activities allow to monitor the qualitative performance of the process, but they are not sufficient nor detailed enough to show a repeatable way for monitoring. For instance, the Business Function element *Receive and check samples* is linked to the Goal element *Samples integrity ensured*, which is connected to the Requirement *Samples must not be damaged or incomplete*. However, there is no connection to some protocol for samples integrity checking. Note that this is also a limitation that was considered for another requirement, showing the natural link between requirements from Indicators defined for the same Process Attribute. Finally, the F - Fully Achieved rating has been given to two requirements from the Indicator PA 2.1.e that refer to the definition and assignation of responsibilities of the process (one requirement is regarding definition and the other one regarding assignation). The responsibilities of the process are considered as well separated and defined, considering the presence of the following three Business Role elements in the model: *Samples provider*; *Samples management*; and *Samples user*, with each Business Function element of the model assigned to one of these roles satisfactorily.

Regarding PA 2.2, results show a more mature process when the Indicators of this PA are considered. Most of the requirements have a rating of L - Largely Achieved (a total of eight) with the two requirements of indicator PA 2.2.a achieving this rating. Moreover, the rating has also

been given to requirements from PA 2.2.d, which are related to the definition and arrangement of work products and requirements for work products of the process. Requirements related to the identification and definition of work products defined by PA 2.2.c also obtained the same rating. Showing that the assessors consider that work products are appropriately identified and defined in the process, although with some margin of improvement. The N - Not Achieved rating has been given to five requirements. The requirements are related to the definition of requirements to document and control of work products, and with the adjustment of work products to fit the arrangements. Finally, one requirement addressing the correct documentation of work products of the process achieved the F - Fully Achieved rating.

Similarly to the analysis of BPMN, the ratings given to the atomic requirements were aggregated to obtain the final ratings for each Indicator. The same aggregation method that has been used in the previous section was used. The final results for the Indicators of PAs 2.1 and 2.2 are shown in Table 5.8.

Indicator	Rating
2.1.a. Objectives for the performance of the process are identified.	P
2.1.b. Performance of the process is planned.	P
2.1.c. Performance of the process is monitored.	N
2.1.d. Performance of the process is adjusted to meet plans.	P
2.1.e. Responsibilities and authorities for performing the process are defined, assigned and communicated.	P
2.1.h Interfaces between the involved parties are managed to ensure both effective communication and clear assignment of responsibility.	N
2.2.a. Requirements for the work products of the process are defined.	L
2.2.b. Requirements for documentation and control of the work products are defined.	P
2.2.c. Work products are appropriately identified, documented, and controlled.	L
2.2.d. Work products are reviewed in accordance with planned arrangements and adjusted as necessary to meet requirements.	P

Table 5.8: Ratings given to the Indicators of the Process Attributes 2.1 and 2.2 of the ISO/IEC 33020 that are visible through ArchiMate models, for the Samples Management Process.

5.5.3 Discussion

The case study presented in this section shows that the methodology proposed in this work can be applied to analyse concrete enterprise models to define the achievement of requirements defined by some assessment framework. This shows that the method is applicable not only for an *a priori* analysis of modelling languages but also during actual enterprise assessments. The goals, questions, metrics and requirements extracted from the ISO/IEC 332020 in Section 5.4 that were initially defined for analysing modelling languages were re-used for the analysis of a concrete enterprise model. The point of divergence (when compared to the analysis of a modelling language), is that the existence of concrete elements answering the EO or EO interaction must be checked, followed by a qualitative analysis of the assessor to define the extent to which the elements answer the atomic requirement. Moreover, it is worth mentioning that Enterprise Objects may not be only considered as modelling elements for the analysis of enterprise models, but real-world entities that can be registered by the assessors. This allows to consider not only enterprise models but other types of assessment evidence. The case study also confirmed what has been concluded in Section 5.4: it is possible to visualise more information about the

process using ArchiMate. Therefore, since there is more visible information in ArchiMate, more requirements are able to be assessed and the assessment result is more complete, when compared to that obtained by analysing the BPMN model. However, note that this would not hold true if the ArchiMate model is designed only considering functional aspects of a process or using elements from a single layer. Moreover, there could be several models, each defining aspects of the process from distinct viewpoints. In such case, the analysis must be focused on all models to yield a relevant and complete result.

5.6 The Enterprise Models Analysis Approach in the Design Science Context

In the scope of the DSR methodology, the approach introduced in this chapter (considered as a DSR artefact) has as main requirement, within the **Relevance Cycle**, the capability to identify the elements of enterprise models that are able to answer the assessment requirements defined by an assessment framework. For this purpose, the approach has been devised through a set of iterations within the **Design Cycle** that allowed to refine it so as to reach enough robustness to provide satisfactory results. The construction of the approach was based on combining different methodologies related to requirements engineering and enterprise modelling language assessment. Moreover, the validation introduced in Section 5.4 was partially performed more than once in order to expose and resolve possible drawbacks that were noticeable only during the application of the proposed approach. Therefore, the information presented in that section represents the results of the application of the final version of the approach. Finally, considering the **Rigour Cycle**, documents, papers, international standards and expertise regarding enterprise assessment and enterprise modelling were considered as part of the knowledge base, as well as all relevant information addressing requirements engineering and the Goal Question Metric paradigm.

5.7 Conclusion

This chapter introduced a methodology to analyse the suitability of the elements defined by enterprise modelling languages and enterprise models to support organisational assessments. The approach is based on defining a list of atomic requirements, decomposed from an assessment framework, that must be matched with Enterprise Objects that are represented by the modelling language that is the object of analysis, thus providing a link between requirements and modelling elements. The decomposition of the requirements is based on the Pseudo-Requirement Graph approach whilst the matching of requirements with the modelling language elements is done through the Goal Question Metric paradigm. The proposed approach can be used as a method for allowing assessors to understand (*a priori*) which requirements from an assessment framework can be analysed using the enterprise models that exist in an organisation. The method can also be applied to perform assessments, in which concrete enterprise models are analysed to define the conformance of the assessed entity to some requirements.

To test the suitability of the approach, a method validation was performed. It was focused on evaluating the suitability of the BPMN and ArchiMate modelling languages to answer requirements defined by the ISO/IEC 33020 international standard, an assessment framework for Process Capability assessment. The evaluation has been done considering the first two Process Attributes defined in the standard. Results showed that, for the Process Attribute that is focused on functional aspects of the assessed process (PA 1.1), BPMN is a satisfactory alternative to provide a view on those aspects, whilst it only partially supports the PAs that are focused

on more non-functional aspects (PAs 2.1 and 2.2). Some of these aspects, on the other hand, can be addressed through ArchiMate models, although with some limitations, since it provides a larger list of elements to describe organisational aspects. A case study based on analysing concrete BPMN and ArchiMate models to assess the PAs of Capability Level 2 of a business process, considering Indicators from the ISO/IEC 33020 standard, was also performed. Results showed that the methodology does not only allow to analyse modelling languages but it is also suitable for the analysis of specific enterprise models. Moreover, since the information regarding which requirements were able to be answered with these modelling languages (defined through the method validation) was already defined, it was possible to only concentrate on concrete Indicators that were analysable using each language, thus reducing the necessary effort to perform the assessment. Note that this characteristic is also useful for the development of smart systems in charge of automating tasks of an assessment, allowing system architects and developers to focus on implementing system components to treat only the Indicators that can be answered by the enterprise models to be used as evidence.

Future work will focus on testing the capability of the proposed approach to tackle the combination of models stemming from various modelling languages. Moreover, the list of EOs/EOIs and their link to the atomic requirements obtained after the application of our approach could also be used as information source to build more complete models in order to better reflect reality. Likewise, another future research path will aim at providing means to automate the assessment process, considering that the Pseudo-Requirement Graph approach for Requirements Decomposition already allows to generate a list of formally defined atomic requirements. Note that the automation must not be only focused on the EO and requirements matching task but also on evaluating the supported requirements, using enterprise models as point of reference, to provide assessment results, as shown in the case study. Automation in this context can be a relevant step towards the development of smart assessment methods, which could be able to consider enterprise models as assessment evidence and provide assessment results. Therefore, efforts could be directed towards the development of smart systems able to consume enterprise models and process them automatically. This is explored in detail in the next chapter, in which an approach to automate the analysis of business process models using a data-driven AI method is presented.

6

An Approach to Evaluate Business Process Capability Using Process Models

Contents

6.1	Introduction	109
6.2	Previous Attempts to Evaluate Process Models	110
6.3	GCN Approach for Business Process Models Assessment	111
6.3.1	Process Model Generator	112
6.3.2	Process Model Classifier	114
6.3.3	GCN Deployment	115
6.4	GCN Approach Experiments and Results	116
6.4.1	System Setup	116
6.4.2	Dataset	116
6.4.3	Results	116
6.5	The GCN Approach in the Design Science Context	118
6.6	Conclusion	118

6.1 Introduction

One of the possible alternatives to describe business processes in organisations is through the use of enterprise models. Specifically, business process models are enterprise models that describe the elements that take part of the execution of a process. Their purposes are many-fold, including documentation, analysis, control, and improvement. There are several modelling languages that are used to represent them such as Petri Nets [290], UML Activity diagrams [234], Object-Centric Behavioral Constraint (OCBC) [291], or Business Process Model and Notation (BPMN) diagrams [166].

As presented in the previous chapter, business process models are subject to be used as assessment evidence when performing process capability assessment. Note that evidence, in this context, represents any type of data reflecting the state of an assessed entity [16]. If there is an available process model describing the activities and elements of the business process, an

assessment team could examine it in order to check if the process complies with the requirements defined by a specific assessment framework. However, manually analysing the models is resource-demanding. It requires time and effort of the assessors that are part of the assessment team, and it is also prone to possible human errors. For this reason, reducing the amount of effort that is required to perform an appraisal could be valuable to improve the overall assessment process. Specifically, methods to automatically analyse process models so as to check if a process complies to a set of requirements could serve for such purpose. The work presented in this chapter aims at addressing this issue by introducing an approach to automatically evaluate the correctness of process models considering the order of their activities and their content, which are required to achieve the objectives of the process. In this context, a process is considered correct if it achieves its expected goals and produces the required outputs. The method is entirely data-driven, which implies that no human intervention is required to devise rules or heuristics that the system will use, on the opposite to model-driven approaches. The main idea of the work is to support assessments from Capability Level 1 of the ISO/IEC 33020 framework [20], which has a single requirement related to the accomplishment of the main objective of the business process. Note that the work presented in Chapter 5, more specifically the results presented in Figure 5.6 of that chapter, showed that this capability level can be answered through the use of modelling languages that allow to define a set of interrelated activities such as BPMN.

The approach proposed in this chapter is based on the use of a Graph Neural Network (GNN) to act as classifier of process models, which are pre-processed to be used as input graph data. A GNN is a Deep Learning (DL) approach [13]. DL methods allow to discover intricate structure in large datasets. Since a large set of samples is required to train DL models so as to obtain satisfactory results, a method to simulate process models is also introduced in order to synthesise samples that are used as part of the dataset to train and test the GNN. The work presented in this chapter is focused on answering the following Research Question:

- Is a Graph Neural Network-based approach suitable to evaluate business process models for supporting capability assessment?

It is worth mentioning that, in the context of the Smart Assessment Framework (SAF), the GNN in charge of evaluating the business process models and the mechanisms in charge of tuning the network so as to provide accurate results are considered as the Knowledge and Learner components of the SAF, respectively. This is analogous to the Long Short-Term Memory (LSTM) network used for treating text evidence, presented in Chapter 4.

This chapter is organised as follows. Section 6.2 describes previous attempts to perform evaluations of business process models. Section 6.3 presents the proposed GNN-based approach, whilst a description of the experiments performed to validate the approach is detailed in Section 6.4. Section 6.5 describes the proposed approach as a Design Science artefact. Finally, Section 6.6 presents the conclusions of the work and enumerates future research perspectives.

6.2 Previous Attempts to Evaluate Process Models

This section presents an overview of the approaches that were previously introduced in the literature to automatically evaluate business process models from a qualitative perspective. The following trends are identified: hand-crafted methods, which require manually defined heuristics to treat the process models for the assessment; and data-driven approaches, which rely on the automatic extraction of knowledge from data so as to analyse and evaluate the models.

From the **hand-crafted** methods perspective, the approach introduced in [103] allows to study the flow of private information in process models that are defined using BPMN diagrams.

The authors extend BPMN so as to include activities related to privacy-enhancing technologies in order to provide stakeholders awareness regarding potential privacy risks. On the other hand, the work presented by [104] proposed the use of semantic technologies to automatically analyse Enterprise Architecture (EA) models to support maturity assessment. The approach is based on the design of ontologies to represent EAs through templates and Description Logic queries to perform the evaluation. A case study focused on Risk Assessment was introduced to validate the method. Likewise, [105] presented an approach for process improvement at design time through the simulation of the workflow from a visual perspective, in which the performance output of each simulated path of the workflow is shown. The method was validated through a case study focused on simulations of a quality documents assessment process. On the other hand, note that a certain process model can also be compared to another one (a standard process for instance) for an automatic evaluation. In this context, several metrics have been introduced in the literature [292] to aid at measuring similarity between process models. The work by [293], for instance, presented a set of metrics considering aspects such as behaviour coded in the models or the meaning of their elements. Moreover, ontological approaches based on ontology-matching [294–296] could also be applied to establish comparisons between process models, given that the models are represented through ontologies. For instance, the work presented by [106] performed fraud assessment in business process models using such type of techniques.

Among the **data-driven** approaches, Conformance Checking, a subset of techniques of Process Mining [39], which is based on automatically pre-processing event logs from systems so as to obtain process models, allows to diagnose and quantify the level of conformance of a process model to the actual observed behaviour of the process, represented through process logs [297]. The main limitation of these techniques is that event logs are required to perform conformance checking, although some works have been recently focused on the extraction of process models from other sources such as emails [298, 299]. If only event logs are considered, generally the elements of the process models contain limited information and more complex models are not taken into account. Other data-driven approaches are focused on the use of DL techniques to analyse process models, the work by [300], for instance, employed LSTM [260], a specific type of Recurrent Neural Network, to classify labels contained in business process models. The method is specifically focused on classifying the labels attached to the nodes of a process model. This was done by representing the labels through numerical vectors known as word embeddings and feed them to a LSTM that is trained as a classifier. The authors presented a case study that consisted in training and evaluating the approach on labels of German process models.

Considering the works presented in this section, there seems to exist a lack of works addressing an automatic qualitative analysis (the base for capability and maturity assessment) of business process models from a data-driven perspective. Moreover, there seems to exist an insufficiency of DL-based approaches dealing with the treatment of process models considering both, the text of the labels attached to their nodes and the way those nodes are linked in order to perform qualitative assessment. This chapter introduces an approach to tackle these issues, which is described in detail in the following section.

6.3 GCN Approach for Business Process Models Assessment

The work presented in this chapter aims at performing the automatic evaluation of business process models using a data-driven classification method. Specifically, a GNN is used to act as classifier of process models that are used as input graph data. The task is tackled as a two-class classification problem, in which a process model that fulfils the expected requirements to reach

a certain quality level belongs to the positive class and a model that does not reach the level belongs to the negative class. The proposed methodology follows a set of steps described in Figure 6.1. A BPMN diagram [166] is used to describe the process.

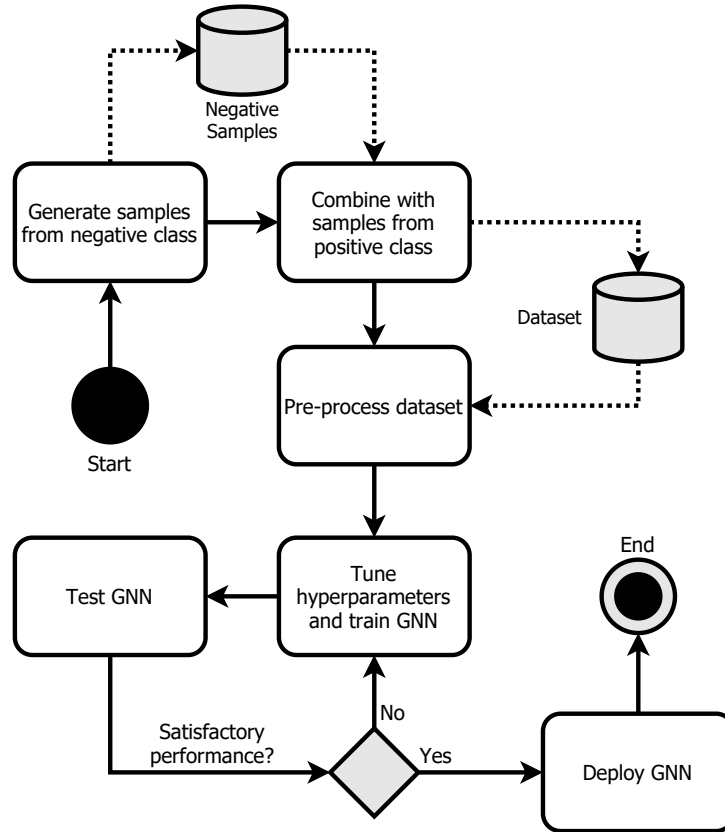


Figure 6.1: Overview of the methodology followed in this work.

The first step of the process is to define the data that will be used for training the network. This is done by synthetically generating process models belonging to the negative class following the approach presented in Section 6.3.1. The generated models are then combined with a set of process models from the positive class that are manually designed by experts. After, the models that are part of the dataset are pre-processed so as to have the format and data structure that could be suitable to be processed by a GNN. Then, elements such as the architecture of the network and the training hyperparameters are empirically defined, and the network is trained using a hold-out dataset split. Details of the architecture of the network used in this chapter are presented in Section 6.3.2. Finally, the classification capability of the network is evaluated and if it is satisfactory, the GNN is deployed to support capability assessment in a production setting.

6.3.1 Process Model Generator

Unlike raw data collected from sensors or unstructured text data (defined in emails, documents, reports, etc.) that are generated on a daily basis in organisations. Enterprise models such as business process models are more difficult to obtain. In general, a certain amount of time and effort is required from humans to define the elements of the model and their relationships. This is due to that they must follow specific rules for the correct design of the model in terms of syntactic and semantic, and they need to have a deep understanding of the entity they are

representing through the model. Since the problem addressed in this work is viewed from a data-driven classification perspective, there is a need of data to train the classifier. Therefore, an approach to synthetically generate a dataset is devised. The synthetic dataset is composed of business process models that are incorrect when considering quality of the process, thus samples from the negative class are generated. As for the positive class, the process models are manually defined by experts to ensure that they are correct. The pipeline to generate the process models is presented in Algorithm 1.

Algorithm 1 Algorithm to generate graphs belonging to the negative class.

```

1: dataset = [ ]
2: input_data = [element0, element1, ... , elementn]
3: i = 0
4: while i < ngraphs do
5:   init Graphi
6:   Graphi.add_node(start_node)
7:   shuffled = shuffle(input_data, prob)
8:   nnodes = random(0, n)
9:   j = 0
10:  while j < nnodes do
11:    s = word_net_to_sentence(shuffled[j])
12:    node = sentences_to_indices(s, dict)
13:    Graphi.add_node(node)
14:    j = j + 1
15:  end while
16:  Graphi.add_node(end_node)
17:  dataset.append(Graphi)
18:  i = i + 1
19: end while

```

The main input of the algorithm is the *input_data* vector, which contains *n* elements of a process model comprising activities, business objects, events, etc., that can be used to define the process model for a given case. Each element of *input_data* is a tuple (*description*, *type*). For instance, for an activity element, the tuple would be as follows: (“Return product to customer”, “activity”), whilst for a message element, a possible tuple would be: (“Rejection letter”, “data object”). Besides elements of *input_data*, two tuples defining the “Start” and “End” activities are pre-defined and used in the algorithm.

The algorithm initiates with a loop that allows to generate a certain number *ngraphs* of process models. At each time step, a graph *Graph*_{*i*} is defined, a Start node *start_node* is added to the graph, the elements of *input_data* are shuffled with a certain probability value defined by the user, and the number of elements *nnodes* that are going to be used from *input_data* is randomly defined. This is done to generate possible incomplete processes. A second loop is then applied to generate a *node* using each element of *input_data* and then added to *Graph*_{*i*}. Note that there is a pre-processing step to treat the text of each element before creating and adding the node to the graph. For such purpose, the following functions are used: *word_net_to_sentence* and *sentences_to_indices*.

The *word_net_to_sentence*(*new_node*) function receives a tuple *new_node* containing a pair [*description*, *type*] and converts each word of *description* into its synonym or antonym

defined in WordNet¹⁰ [301] with a probability $wnetprob$. Stop-words are not considered for conversion. The value $wnetprob$ is defined by the user and it is generally low (equal or less than 0.2) specially for the antonyms. Moreover, since there are more than one possible synonyms or antonyms per word defined by WordNet, the new word is also randomly selected from the possible options. The idea behind the function is to provide variety to the dataset by creating nodes with heterogeneous content. The output of the function is an array s . On the other hand, the $sentences_to_indices(s)$ function converts the array s containing words (strings) into an array of indices corresponding to each word in a given dictionary. The array is also padded with zeros to have the dimension of the longest possible sentence. The output of the function is the $node$ to be added to the $Graph_i$. Note that $node$ also includes the information addressing the type of element and not only the array of integers.

The final steps of the algorithm consist of adding the end_node to the $Graph_i$, including edges between nodes to connect them, and appending the graph to $dataset$, which is a list of graphs representing the final output of the algorithm. A label from the negative class is associated to each graph of the dataset, showing that all models are incorrect. Note that, in the context of this work, an incorrect process means that the elements of the process model contain incorrect textual information or that they are connected in the wrong order.

6.3.2 Process Model Classifier

GNNs are deep learning models that aim at solving graph-related tasks in an end-to-end manner [302]. Specifically, a Graph Convolutional Neural Network (GCN) is used in this work. A GCN is a type of GNN in which the local filter parameters are shared over all locations in a graph [303].

In this work, a business process model is treated as a graph $G = (V, E)$ where V is a set of nodes and E is a set of edges connecting the nodes. Each node $v_i \in V$ has a feature vector $f_i \in \mathbb{N}^d$ of dimension d that is composed of a set of indexes from words in a dictionary. The feature vector represents a sentence describing the content of an element of the process model. On the other hand, $e_{ij} = (v_i, v_j) \in E$ is an edge pointing from v_i to v_j . This work does not consider features for edges given the nature of the problem, although they can be used in other settings. Each graph G_k , of a dataset composed of N graphs, is used as input of the GNN. Figure 6.2 presents an example of a visual representation of a process model using the graph structure and the architecture of the network.

The neural network is composed of an Embedding layer that transforms a feature vector f_i of a node i of G_k into an embedding vector x_i . The layer uses weights from the GloVe model [264] pre-trained with Gigaword5¹¹ and Wikipedia text data. This layer is not trainable, thus it is devised to act as a pre-processing step of the input graph.

The graph is then fed to a set of graph convolutional layers. The layers compute a new representation for each node through an aggregation process that considers information from its neighbours. The input of the first layer is composed of embedding vectors, given by the Embedding layer, summarised in a feature matrix \mathbf{X} of size $M \times D$ where M is the number of nodes and D is the dimension of embedding vectors, and an adjacency matrix \mathbf{A} which is a matrix representation of the structure of the graph. A convolutional layer of the network can be represented through the formulation presented in Equation 6.1, defined in [304].

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)}), \quad (6.1)$$

¹⁰wordnet.princeton.edu

¹¹catalog ldc.upenn.edu/LDC2011T07

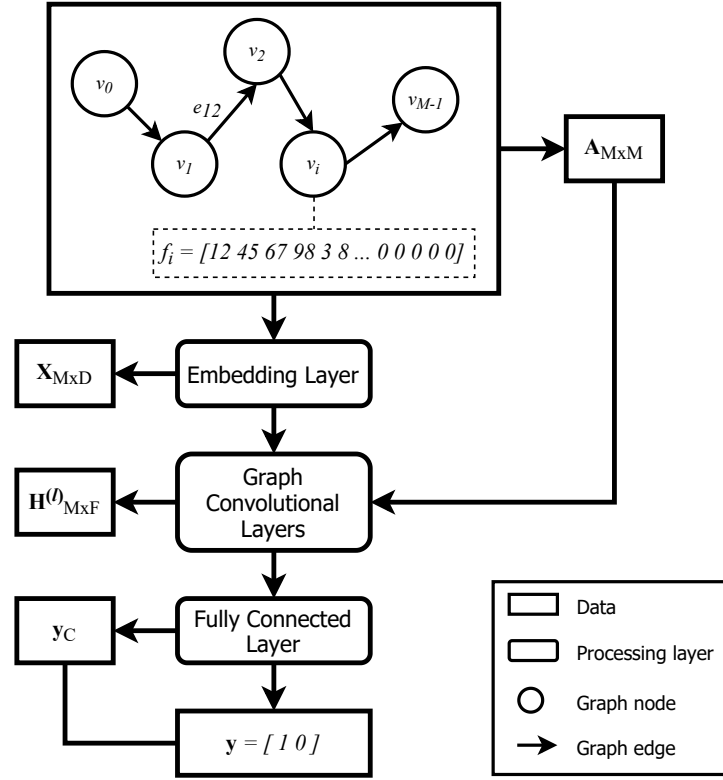


Figure 6.2: Example of a visual representation of an input process model and the network architecture.

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_M$ is the adjacency matrix of G_k with self-connections; \mathbf{I}_M is the identity matrix; $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$; and $\mathbf{W}^{(l)}$ is a weight matrix of the l -th layer. Moreover, $\sigma(\cdot)$ is an activation function, usually $ReLU(\cdot) = \max(0, \cdot)$; $\mathbf{H}^{(1)} \in \mathbb{R}^{M \times F}$ with $\mathbf{H}^{(0)} = \mathbf{X}$. In this work, $l = 2$.

A pooling operation is applied on the output of the final convolutional layer to combine the features of all nodes to obtain a final graph-level representation. In this work, we apply the average of all node features for such purpose. The output of the pooling operation, a vector of size F , is then fed to a Fully Connected layer, that acts as the classifier, which outputs a vector \mathbf{y} of size C , equal to the number of possible classes for the graph. In this work, $C = 2$ given that the graphs can be either correct or incorrect. The output of a fully connected layer is given by a matrix multiplication between the input of the layer and a weight matrix added to a bias term.

To train the GCN in this work, the Cross-Entropy is used as loss function L , which was defined in Equation 4.1, introduced in Chapter 4. The algorithm used to optimise the parameters of the network is Adam [274], which is a stochastic gradient descent-based optimisation method. The algorithm is based on computing individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients; hence the name Adam, derived from adaptive moment estimation [274].

6.3.3 GCN Deployment

Once the network has achieved a satisfactory performance after the training and testing phase, it is ready to be deployed in production. Often, an acceptable performance is decided by comparing

its results to those provided by a baseline approach. Specifically, the approach presented in this work can be used as a component of a system devised by following the architecture defined in the SAF introduced in Chapter 3. In this sense, the network acts as the Knowledge component that has previously learned (using a Learner component) to evaluate process capability using process models as input assessment evidence.

6.4 GCN Approach Experiments and Results

This section presents the experiments performed to validate the proposed approach. First, the setup of the system used to perform the experiments is presented. Then, the main aspects of the dataset are detailed. Finally, the results obtained after applying the approach on the dataset is described.

6.4.1 System Setup

A server with a processor Intel Core i9 with 8 cores at 2.3 GHz and 16 GB 2400 MHz DDR4 was used to implement, train and test the GCN. The server runs macOS Big Sur, version 11.2.1. The following Python libraries were used: NumPy, version 1.19.1, Torch, version 1.6.0, Networkx, version 2.4, Scikit-learn, version 0.23.2, and DGL, version 0.4.

6.4.2 Dataset

In this work, the focus was set on classifying business process models composed entirely of activity elements. A total number of 91 incorrect models (class 0) were simulated using the method described in Section 6.3.1; and a group of domain experts designed 10 correct models (class 1). Note that, in this context, correctness is given by an appropriate content of the activity elements and their proper order. The models describe a process that comprises activities of reception, treatment, destruction, and returning of different types of chemical samples that are used in an organisation. This is the same scenario that was used for the case studies in Chapters 4 and 5. Figure 6.3 presents an example of a correct or compliant process model. It also shows the content of each node composed of the description, type, and identifier.

The dataset has been shuffled and a hold-out validation strategy was used for training and testing the GCN. The distribution of the samples was defined as follows: 70% for training and the remaining 30% for testing.

6.4.3 Results

The parameters of the network were optimised using the Adam [274] algorithm, with an initial learning rate equal to 0.0005. The training process was performed through 60 epochs. The GCN is composed of two graph convolutional layers with 128 processing units for each layer, with input feature vectors of size equal to 500 for each node after passing through the embedding layer, considering a maximum of 10 nodes per graph. The same number of neurons is used for the fully connected layer, which outputs two-dimensional vector. The evolution of the loss for the train and test splits of the dataset during training is shown in Figure 6.4.

A baseline was defined in order to have a point of reference to evaluate the results of the proposed approach. The baseline was based on representing the content of the nodes of each process model through a matrix of word token counts to be used as input features to be fed to a classifier. The Support Vector Machines (SVM) algorithm [305] was used as the classifier given

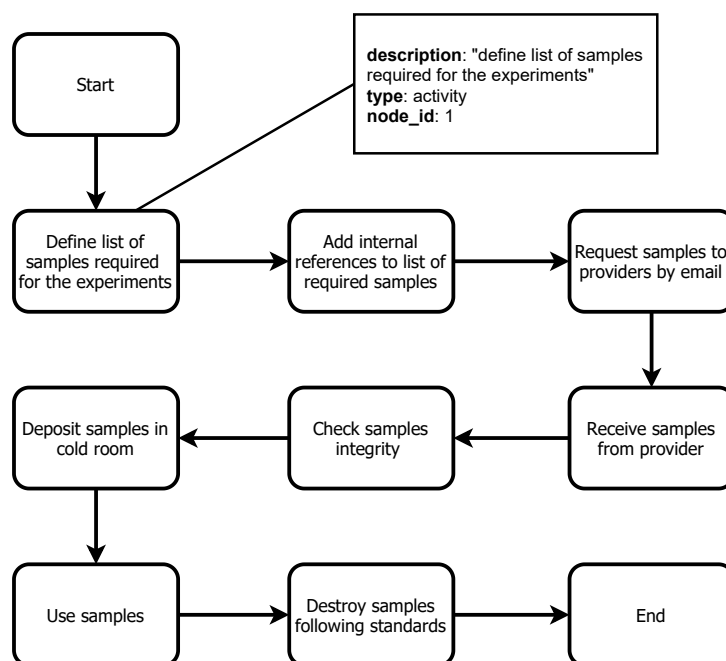


Figure 6.3: Example of a correct process model of the dataset.

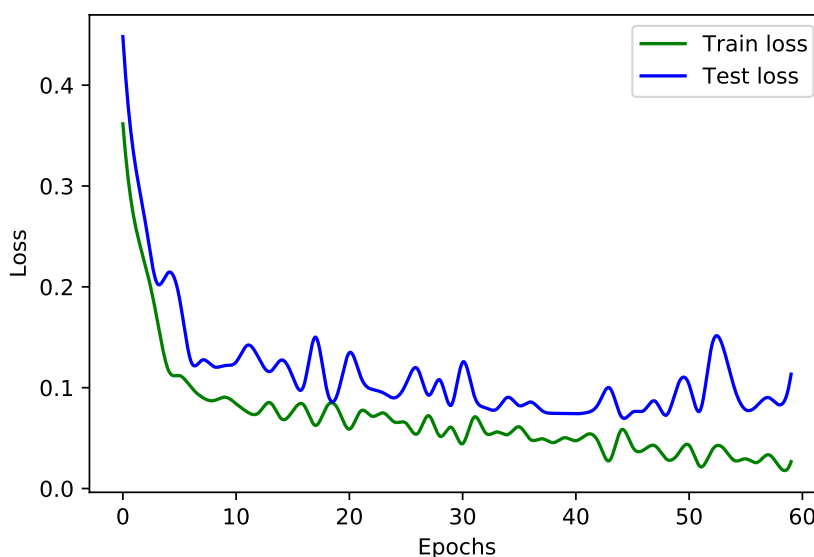


Figure 6.4: Evolution of the loss of the network for the train and test sets throughout the epochs.

its simplicity and ease of implementation. The Scikit-learn library was used to implement and test the baseline with the default values for the parameters of each function provided by the library. Results show that, in terms of raw accuracy, the GCN achieved 94% whilst the baseline got 90%. The approach introduced in this work is better by a small margin. However, due to that the dataset is highly imbalanced, other evaluation metrics are more suitable. Table 6.1 shows values of Precision, Recall, and F1 Score achieved by both approaches.

Results show a better performance from the GCN approach considering all metrics for both classes, except for the Recall for Class 0 and Precision for Class 1, which were higher for the

Table 6.1: Classification results obtained by the GCN and the Baseline for each class of the train split of the dataset. MAVG: Macro Average. WAVG: Weighted Average.

	GCN Approach			Baseline Approach		
	Precision	Recall	F1	Precision	Recall	F1
Class 0	0.96	0.96	0.96	0.90	1.00	0.95
Class 1	0.75	0.75	0.75	1.00	0.25	0.40
MAVG	0.86	0.86	0.86	0.95	0.62	0.67
WAVG	0.94	0.94	0.94	0.91	0.90	0.88

Baseline. Note that the Recall represents the true positive rate or Sensitivity, whilst the Precision is also known as the positive predictive value. These results are due to that it overfits the data of the majority class (Class 0), since it classified only one instance as part of the Class 1, which was correct compared to the original label of that instance. However, all other instances of Class 1 from the ground-truth were incorrectly classified. This is visible if the Macro and Weighted Average of the metrics are considered, both show a better performance from the GCN approach, except when considering the Precision (for the Macro Average) given by the high Precision value for Class 1. Moreover, the values for the F1 Score, which is score given by a function that relates Precision and Recall, for each method show that the GCN provides a more robust classification capability.

6.5 The GCN Approach in the Design Science Context

The GCN approach is another Design Science artefact intended to evaluate business process models to support capability assessments. Indeed, this is the main requirement within the **Relevance Cycle**. This requirement is fulfilled through the design, implementation, training and testing of the GCN as part of the **Design Cycle**. The two main challenges during this cycle were the configuration of the architecture of the GCN and its hyperparameters for the training phase, and the definition of the method to generate the synthetic data to be used for training and testing. The evaluation was done using the hold-out validation method for DL models with typical classification metrics such as Precision, Recall and F1 score. Finally, in the **Rigour Cycle**, the knowledge base used for the work was composed of the literature addressing GNNs and GCNs, the literature addressing business process models design and assessment, as well as the documentation of the DL framework and the diverse libraries that were used to synthesise the process models.

6.6 Conclusion

This chapter presented a Graph Convolutional Neural Network-based approach to perform capability assessment using business process models as assessment evidence. The approach is based on pre-processing the business process models and use them as input graph data for the neural network. The task was tackled as a two-class classification problem, in which the process models could belong to two different categories: correct and incorrect, in terms of the content of their elements and their order. A method to synthesise incorrect process models was also introduced. This method was applied in order to synthesise samples that were afterwards used as part of the dataset employed for training and testing the network. The resulting network is intended

to be used as part of a smart system devised using the SAF, specifically acting as a knowledge component able to provide partial ratings to a process when enterprise models are used as evidence. Moreover, the modular nature of the software tool introduced in Chapter 4 allows to easily integrate the trained GCN as another instance of the Knowledge component of that tool. This is a possible path to be followed in a future work. To validate the approach, the network was trained and tested using a dataset composed of simple process models describing the management of chemical samples. Part of the dataset was synthesised and the remaining samples were manually designed. The proposed approach achieved better performance compared to a Word Token Count and SVM-based baseline. Future work will focus on the treatment of more complex process models in terms of size and content, considering other modelling elements besides process activities. This also applies to the process model synthesising algorithm, which must be modified to produce more complex models. Moreover, a larger dataset, composed of process models from several domains, could be developed and used. The experimentation with different network architectures could also represent a relevant research path. Finally, when performing process capability assessment, other requirements such as the adjustment of the process to meet plans or the presence of monitoring activities must be analysed. Future research initiatives could tackle the analysis of such type of requirements.

General Conclusion

This thesis introduced the Smart Assessment Framework (SAF), a conceptual model grounded on concepts addressing enterprise assessment and the capabilities of smart systems. This framework was devised targeting to achieve the main objective of this work: **define a framework to perform smart assessments of enterprise entities**, which aimed at providing an answer to the main Research Question (RQ):

What are the elements that compose a smart enterprise assessment framework?

Specifically, the design of the SAF was the first specific objective of the work: *design a framework to perform smart assessments of enterprise entities*. The conceptual model, introduced in **Chapter 3**, graphically presents the connection between the different components of a smart assessment grouped into three layers: Management Goals, The Assessment Process, and Application Services. It was designed using the ArchiMate enterprise architecture modelling language. A meta-model defined using a Unified Modelling Language (UML) Class Diagram has also been introduced as an initial step towards a final definition of the SAF. The meta-model shows a different view of a smart assessment and it can be considered as complementary to the SAF. The SAF and the meta-model have been devised following the Design Science Research methodology. In this sense, the main sources of knowledge for the development of these two artefacts include the characteristics of smart systems, obtained through a systematic literature review and the application of natural language processing techniques that are described in **Chapter 2**, and a set of elements related to assessments in enterprises such as international standards, the scientific literature, and the knowledge from assessment specialists. The thesis also presented illustrative instances of the SAF and the meta-model: the first one was related to enterprise interoperability assessment, whilst the second one addressed business process capability assessment for a chemical samples management process using process descriptions defined in text data as assessment evidence.

The second specific objective of this work was related to the validation of the framework: *validate the framework considering different assessment scenarios*. For this purpose, the case study based on the assessment of the chemical samples management process has been considered. In this sense, two main scenarios have been proposed. The difference between those scenarios was given by the evidence that was used. The first scenario was based on the use of process descriptions provided by assessment experts in text format. A hybrid approach focused on the use of neural networks and ontologies was introduced to treat this type of assessment evidence. A software tool acting as a full implementation of the SAF was also developed and presented. The tool uses the hybrid approach as the core of its architecture. The diverse aspects of the hybrid approach, the architecture of the assessment tool, and the technologies that were used during its development were described in **Chapter 4**. The tool was able to effectively provide assessment results using the text evidence as input data, without the need of human participation

during the entire pipeline, except for the input of the process descriptions. The second scenario was focused on the use of enterprise models as assessment evidence. The first step to analyse enterprise models is to understand which assessment requirements are visible in the models. This is ultimately given by the modelling language that was used to design them. In this sense, an approach to perform an analysis to reach such understanding was introduced in this thesis. The method, presented in **Chapter 5**, presents a pipeline that is supported by the use of the Goal Question Metric paradigm, Requirement Decomposition, and Enterprise Object (EO) generation. A case study that consisted in analysing the Business Process Modelling and Notation (BPMN) and ArchiMate modelling languages to support the analysis of requirements defined by the ISO/IEC 33020 Measurement Framework was presented. It has also been shown that the method is applicable to perform manual assessments of enterprise models following a systematic approach. Following this line, **Chapter 6** introduced an approach to perform the automatic evaluation of business processes using process models as input data. It was devised to classify process models that meet a specific requirement related to correctness and those that do not, and it was based on the use of neural networks, word embeddings and synthesised process models that were used as part of a dataset to train and test the network. The method achieved a better classification performance when compared to a baseline approach.

The satisfactory results obtained after implementing the SAF for the scenarios described before allow to support the hypothesis of this thesis: **a conceptual framework grounded on the concepts of smart systems can provide efficient support for automated assessments in enterprises**. This thesis is intended to serve as a source of contributions from two perspectives: conceptual and practical. The SAF is composed of a hand-crafted conceptual structure that allows its elements to be instantiated to define assessment systems to be used in different contexts taking advantages of smart capabilities. Likewise, it provides detailed descriptions of those elements and their relationships, so as to provide clear guidelines for them to be further instantiated. From the practical perspective, the implementations of the SAF (for text and enterprise model processing) introduced in this thesis are fully functional and they can be directly used in industrial scenarios to perform assessments. Moreover, they can be used in an academic context, given that they are also devised to be exploited as founding grounds for the development of other approaches that can be built as expansions or complements of the original implementations. Indeed, several research possibilities have been opened for each SAF implementation considering the existence of certain gaps that have been identified.

Future research perspectives are various and they vary according to which aspect of this thesis is going to be addressed. The possible research initiatives were grouped considering five paths: the SAF and its structure, smart systems and their capabilities, smart text evidence processing for enterprise assessment, enterprise models analysis, smart enterprise models evidence processing, and some scenarios considering current technological trends. The rest of this chapter presents some limitations of the thesis and possible approaches that could be followed to close those gaps considering the paths that were enumerated before.

The Smart Assessment Framework

The SAF has reached a stable version after the iterations that were performed when following the three-cycle view of the Design Science Research method. The framework evolved the Smart Assessment meta-model defined in UML to the conceptual model devised using ArchiMate. Future work could focus on further refining the model, including other concepts that may not have been addressed in this work, or that may emerge during the following years. Moreover, the models can be described using ontologies or other ways to represent the concepts of the SAF and

its instances in a machine-readable manner. Likewise, other illustrative or concrete case studies could be defined for experimentation with SAF, besides the two illustrative cases that have been presented in Chapter 3. Finally, the elements of the framework could be linked and related to concepts described in other meta-models, frameworks or formal definitions describing current state-of-art concepts, approaches, technologies, and systems such as Cyber Physical Systems and Cyber Physical Social Systems, Digital Twins, Blockchain-based systems, or Privacy-enhancing technologies.

Smart Systems

The work addressing the definition of smart systems and their capabilities, presented in Chapter 2, described a fully data-driven approach to isolate the characteristics of smart systems. This means that the characteristics were quantitatively extracted from existing data (definitions in the literature) using some algorithms. However, there could exist capabilities that may not have quantitative relevance (in terms of word count or other metrics) but they could have a high conceptual importance. To address this issue, future research could target the use of data-driven approaches and a more traditional evaluation of papers, articles, and definitions of smart systems that can be found in the literature, in order to devise a complete view of such type of systems. On the other hand, most of the capabilities of smart systems described in the chapter can be considered as atomic such as Reasoning, Learning, Perception, etc. Future work could be focused on the pursuit of complex or emergent characteristics that appear due to the interaction between systems. Finally, a meta-model describing the elements of a smart system was introduced in the chapter, which could be instantiated to demonstrate its applicability. Moreover, the meta-model can be translated into an ontology so as to have a machine-readable representation of the concepts and their relationships. This will allow to easily instantiate individuals, use queries to extract information from the model, and use inference engines to perform reasoning.

Smart text evidence processing

The hybrid approach to process text evidence, introduced in Chapter 4, took advantage of the capabilities of neural networks and ontologies to support business process capability assessment. However, other enterprise entities and aspects may be addressed in future work, including other assessment frameworks and standards. Moreover, different sources of textual information besides comments from assessment experts could be used such as documents, emails, regulations, or comments from stakeholders. Likewise, novel Deep Learning architectures for text processing could be applied such as the state-of-art Transformer model [306]. On the other hand, regarding the software tool that uses the hybrid approach as the core of its architecture, future work could focus on extending the prototype presented in this thesis with the inclusion of features such as profile selection, availability of other assessment frameworks, possibility to use other types of assessment evidence, direct connection to sensors or information sources acting as implementations of the Perception layer of SAF, etc.

Enterprise models analysis

The analysis of enterprise models to perform assessments in organisations is a challenge from which various paths can be opened to be explored. From the perspective of continuing what has been presented in this thesis, other modelling languages could be analysed using the approach proposed in Chapter 5, with the objective to have a repository of "matches" obtained after analysing modelling languages and assessment frameworks. For this purpose, the most used and

well-known languages and frameworks could be analysed, with strong focus on the standardised ones. Moreover, the method could be further refined in order to evolve towards an automated or semi-automated approach, in which one or more tasks of the pipeline defined in the methodology could be performed by computing systems. Note that the efforts could be focused on the two practical uses of the approach introduced in this thesis: evaluating modelling languages and performing assessments of concrete enterprise models. Other research path that is worth to explore is the use of the same approach to perform the opposite task: instead of analysing modelling languages and match them with the requirements extracted from an assessment framework, the list of requirements could be used as knowledge base to design more complete models.

Smart enterprise models processing

In this thesis, the automatic treatment of business process models with a simple structure composed only of activity elements was addressed. The approach was presented in Chapter 6. Future work could be directed towards the treatment of other types of enterprise models with more complex structures. Other datasets and Deep Learning architectures could also be used to perform experiments. On the other hand, the neural network used in this thesis was specifically devised to support business process capability assessment at the Level 1 considering the ISO/IEC 33020 standard (define if the process achieves its objectives following a correct sequence and content of activities). Future work could focus on training models to assess other requirements from different assessment frameworks to evaluate other enterprise aspects. Likewise, the effects of the combination of model-driven and data-driven artificial intelligence approaches to perform the evaluation of enterprise models could be investigated. Furthermore, a software tool could also be devised to use the approach as its core functionality aiming at developing an artefact that could be used in industrial contexts to support the treatment of real-world problems.

Big Data and Cloud Computing Scenarios

A common characteristic of all case studies and experiments presented in this thesis is the relative small amount of data used to test the proposed approaches. Even the data-driven approach for automatically analysing business process models presented in Chapter 6 used a small dataset in spite that most of the data was synthesised. In this case, the amount of synthesised data was kept small to avoid generating a very imbalanced dataset, since the compliant or correct process models were manually designed. Therefore, future implementations of the SAF should be focused on addressing the treatment of large and heterogeneous datasets, with emphasis on the use of Big Data methodologies and technologies to serve as instances of the components of the framework. Moreover, SAF implementations may also take advantage of cloud computing technologies to enable online access to assessment evidence. A new set of experiments considering Big Data and Cloud Computing scenarios could provide relevant feedback to refine the structure of the SAF and the expected behaviour of future implementations of the framework.

Bibliography

- [1] J. Rifkin, “The third industrial revolution: How the internet, green electricity, and 3-d printing are ushering in a sustainable era of distributed capitalism,” *World Financial Review*, vol. 1, no. 1, pp. 4052–4057, 2012.
- [2] P. K. Muhuri, A. K. Shukla, and A. Abraham, “Industry 4.0: A bibliometric analysis and detailed overview,” *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 218–235, 2019.
- [3] C. Ebert and C. H. C. Duarte, “Digital transformation,” *IEEE Softw.*, vol. 35, no. 4, pp. 16–21, 2018.
- [4] A. Rojko, “Industry 4.0 concept: background and overview,” *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 11, no. 5, pp. 77–90, 2017.
- [5] G. Weichhart, H. Panetto, and A. Molina, “Interoperability in the cyber-physical manufacturing enterprise,” *Annual Reviews in Control*, 2021.
- [6] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [7] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial internet of things: Challenges, opportunities, and directions,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.
- [8] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0,” *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.
- [9] M. Ciorciari and P. Blattner, “Enterprise risk management maturity-level assessment tool,” *Society of Actuaries*, vol. 41, p. 25, 2008.
- [10] G. d. S. S. Leal, W. Guedria, and H. Panetto, “An ontology for interoperability assessment: A systemic approach,” *Journal of Industrial Information Integration*, vol. 16, p. 100100, 2019.
- [11] A. Van Looy and A. Shafagatova, “Business process performance measurement: a structured literature review of indicators, measures and metrics,” *SpringerPlus*, vol. 5, p. 1797, oct 2016.
- [12] S. Russell and P. Norvig, “Artificial intelligence: a modern approach,” 2002.
- [13] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, p. 436, 2015.

- [14] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.
- [15] L. A. Zadeh, “Fuzzy logic,” *Computer*, vol. 21, no. 4, pp. 83–93, 1988.
- [16] ISO Central Secretary, “ISO/IEC 33001:2015 Information technology — Process assessment — Concepts and terminology,” standard, International Organization for Standardization, Geneva, CH, mar 2015.
- [17] ISO Central Secretary, “ISO/IEC 33002:2015 Information technology — Process assessment — Requirements for performing process assessment,” standard, International Organization for Standardization, Geneva, CH, mar 2015.
- [18] C. P. Team, “CMMI for Development, Version 1.3,” Tech. Rep. CMU/SEI-2010-TR-033, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2010.
- [19] G. O’Regan, “SCAMPI Appraisals,” in *Introduction to Software Process Improvement*, pp. 221–238, London: Springer London, 2011.
- [20] ISO Central Secretary, “ISO/IEC 33020:2015 Information technology — Process assessment — Process measurement framework for assessment of process capability,” standard, International Organization for Standardization, Geneva, CH, mar 2015.
- [21] B. Barafort, B. Di Renzo, and O. Merlan, “Benefits resulting from the combined use of ISO/IEC 15504 with the Information Technology Infrastructure Library (ITIL),” in *International conference on product focused software process improvement*, pp. 314–325, Springer, 2002.
- [22] M. Rohloff, “Advances in business process management implementation based on a maturity assessment and best practice exchange,” *Information Systems and e-Business Management*, vol. 9, pp. 383–403, sep 2011.
- [23] A. Rula and A. Zaveri, “Methodology for assessment of linked data quality,” in *LDQ@ SEMANTICS*, p. 34, Citeseer, 2014.
- [24] W. Guédria, Y. Naudet, and D. Chen, “Maturity model for enterprise interoperability,” *Enterprise Information Systems*, vol. 9, no. 1, pp. 1–28, 2015.
- [25] L. A. Lasrado, R. Vatrapu, and K. N. Andersen, “Maturity models development in IS research: a literature review,” in *IRIS Selected Papers of the Information Systems Research Seminar in Scandinavia*, vol. 6, 2015.
- [26] A. P. Goncalves Filho and P. Waterson, “Maturity models and safety culture: A critical review,” *Safety science*, vol. 105, pp. 192–211, 2018.
- [27] R. H. Lok and A. Walker, “Automated tool support for an emerging international software process assessment standard,” *Proceedings of IEEE International Symposium on Software Engineering Standards*, pp. 25–35, jun 1997.
- [28] W. Wen, Y. Chen, and I. Chen, “A knowledge-based decision support system for measuring enterprise performance,” *Knowledge-Based Systems*, vol. 21, no. 2, pp. 148–163, 2008.
- [29] G. Leal, W. Guédria, H. Panetto, and E. Proper, “Towards a semi-automated tool for interoperability assessment: An ontology-based approach,” 2017.

-
- [30] B. Barafort, A. Shrestha, S. Cortina, and A. Renault, “A software artefact to support standard-based process assessment: Evolution of the TIPA $\text{\textcircled{R}}$ framework in a design science research project,” *Computer Standards and Interfaces*, vol. 60, pp. 37–47, nov 2018.
- [31] M. Rafiei, L. v. Waldthausen, and W. M. van der Aalst, “Supporting confidentiality in process mining using abstraction and encryption,” in *Data-Driven Process Discovery and Analysis*, pp. 101–123, Springer, 2018.
- [32] M. Rafiei and W. M. van der Aalst, “Practical aspect of privacy-preserving data publishing in process mining,” *arXiv preprint arXiv:2009.11542*, 2020.
- [33] X. Zhou, G. Zacharewicz, D. Chen, and D. Chu, “A method for building service process value model based on process mining,” *Applied Sciences*, vol. 10, no. 20, p. 7311, 2020.
- [34] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, “Context-aware computing, learning, and big data in internet of things: a survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 1–27, 2017.
- [35] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M.-C. Shan, “Business process intelligence,” *Computers in industry*, vol. 53, no. 3, pp. 321–343, 2004.
- [36] M. Langer and D. Söffker, “Human guidance and supervision of a manufacturing system for semi-automated production,” in *2011 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pp. 1–6, IEEE, 2011.
- [37] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the internet of things: A survey,” *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 414–454, 2013.
- [38] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggle, “Towards a better understanding of context and context-awareness,” in *International symposium on handheld and ubiquitous computing*, pp. 304–307, Springer, 1999.
- [39] W. Van Der Aalst, *Process mining: discovery, conformance and enhancement of business processes*, vol. 2. Springer, 2011.
- [40] W. Van Der Aalst, A. Adriansyah, A. K. A. De Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. Van Den Brand, R. Brandtjen, J. Buijs, *et al.*, “Process mining manifesto,” in *International conference on business process management*, pp. 169–194, Springer, 2011.
- [41] C. Kung and A. Soelvborg, “Activity modeling and behavior modeling,” in *Proc. of the IFIP WG 8.1 working conference on Information systems design methodologies: improving the practice*, pp. 145–171, 1986.
- [42] J. Bigot and C. Pérez, “Increasing reuse in component models through genericity,” in *International Conference on Software Reuse*, pp. 21–30, Springer, 2009.
- [43] P. Soffer and I. Hadar, “Reusability of conceptual models: The problem of model variations,” in *EMMSAD’03, Proceedings of the Eighth CAiSE/IFIP8. 1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (on CD)*, 2003.
- [44] M. Romero, W. Guédria, H. Panetto, and B. Barafort, “Towards a characterisation of smart systems: A systematic literature review,” *Computers in Industry*, vol. 120, p. 103224, 2020.

- [45] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [46] R. S. Aguilar-Saven, “Business process modelling: Review and framework,” *International Journal of production economics*, vol. 90, no. 2, pp. 129–149, 2004.
- [47] F. Vernadat, “Enterprise modelling: Research review and outlook,” *Computers in Industry*, vol. 122, p. 103265, 2020.
- [48] R. H. Von Alan, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.
- [49] A. R. Hevner, “A three cycle view of design science research,” *Scandinavian journal of information systems*, vol. 19, no. 2, p. 4, 2007.
- [50] C. D. Manning, C. D. Manning, and H. Schütze, *Foundations of statistical natural language processing*. MIT press, 1999.
- [51] Cambridge University Press, *Cambridge online dictionary*. Cambridge University Press, 2008.
- [52] O. Ozcan-Top and O. Demirörs, “A reference model for software agility assessment: AgilityMod,” in *International Conference on Software Process Improvement and Capability Determination*, pp. 145–158, Springer, 2015.
- [53] S. Marshall and G. Mitchell, “An e-learning maturity model,” in *Proceedings of the 19th Annual Conference of the Australian Society for Computers in Learning in Tertiary Education, Auckland, New Zealand*, pp. 8–11, 2002.
- [54] S. Agarwal, S. Steyskal, F. Antunovic, and S. Kirrane, “Legislative compliance assessment: framework, model and gdpr instantiation,” in *Annual Privacy Forum*, pp. 131–149, Springer, 2018.
- [55] O. E. Adali, O. O. Top, and O. Demirors, “Assessment of agility in software organizations with a web-based agility assessment tool,” *Proceedings - 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017*, pp. 88–95, 2017.
- [56] K. Benjamin, C. Cappelli, and G. Santos, “Organizational Transparency Maturity Assessment Method,” in *Proceedings of the 18th Annual International Conference on Digital Government Research, dg.o '17, (New York, NY, USA)*, pp. 477–484, ACM, 2017.
- [57] A. Tarhan, O. Turetken, and F. Ilisulu, “Business Process Maturity Assessment: State of the Art and Key Characteristics,” in *Proceedings - 41st Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2015*, pp. 430–437, 2015.
- [58] F. L. Oliva, “A maturity model for enterprise risk management,” *International Journal of Production Economics*, vol. 173, pp. 66–79, 2016.
- [59] D. Proença and J. Borbinha, “Maturity models for information systems-A state of the art,” *Procedia Computer Science*, vol. 100, pp. 1042–1049, 2016.
- [60] A. Schumacher, S. Erol, and W. Sihn, “A maturity model for assessing Industry 4.0 readiness and maturity of manufacturing enterprises,” *Procedia Cirp*, vol. 52, pp. 161–166, 2016.

-
- [61] A. Cater-Steel, R. Valverde, A. Shrestha, and M. Toleman, "Decision support systems for IT service management," *International Journal of Information and Decision Sciences*, vol. 8, no. 3, p. 284, 2016.
- [62] T. C. Lacerda, C. G. von Wangenheim, and J. C. R. Hauck, "A Mapping Study on Software Process Self-Assessment Methods," *CoRR*, vol. abs/1812.0, 2018.
- [63] G. Grambow, R. Oberhauser, and M. Reichert, "Automated Software Engineering Process Assessment: Supporting Diverse Models using an Ontology," *Int'l Journal on Advances in Software*, vol. 6, no. 1 & 2, pp. 213–224, 2013.
- [64] N. Krivograd, P. Fettke, and P. Loos, "Development of an intelligent maturity model-tool for business process management," *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 3878–3887, 2014.
- [65] A. Giovannini, A. Aubry, H. Panetto, M. Dassisti, and H. El Haouzi, "Ontology-based system for supporting manufacturing sustainability," *Annual Reviews in Control*, vol. 36, no. 2, pp. 309–317, 2012.
- [66] B. Barafort, V. Betry, S. Cortina, M. Picard, A. Renault, M. St-Jean, and O. Valdés, *ITSM Process Assessment Supporting ITIL (TIPA)*. Van Haren, 2009.
- [67] A. K. Sangaiah, O. W. Samuel, X. Li, M. Abdel-Basset, and H. Wang, "Towards an efficient risk assessment in software projects—Fuzzy reinforcement paradigm," *Computers & Electrical Engineering*, vol. 71, pp. 833–846, 2018.
- [68] H. Zhang, Y. Ouzrout, A. Bouras, A. Mazza, and M. M. Savino, "PLM components selection based on a maturity assessment and AHP methodology," in *IFIP International Conference on Product Lifecycle Management*, pp. 439–448, Springer, 2013.
- [69] U. Yudatama and R. Sarno, "Evaluation maturity index and risk management for it governance using Fuzzy AHP and Fuzzy TOPSIS (case Study Bank XYZ)," in *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pp. 323–328, IEEE, 2015.
- [70] J. Pöppelbuß and M. Röglinger, "What makes a useful maturity model? a framework of general design principles for maturity models and its demonstration in business process management.," in *Ecis*, p. 28, 2011.
- [71] J. Becker, R. Knackstedt, and J. Pöppelbuß, "Developing maturity models for IT management," *Business & Information Systems Engineering*, vol. 1, no. 3, pp. 213–222, 2009.
- [72] A. Maier, J. Moultrie, and P. J. Clarkson, "Developing maturity grids for assessing organisational capabilities: Practitioner guidance," in *4th International Conference on Management Consulting: Academy of Management*, 2009.
- [73] T. De Bruin, R. Freeze, U. Kaulkarni, and M. Rosemann, "Understanding the main phases of developing a maturity assessment model," *ACIS 2005 Proceedings*, 2005.
- [74] M. Kohlegger, R. Maier, and S. Thalmann, "Understanding Maturity Models Results of a Structured Content Analysis," 2009.
- [75] J. B. S. D. Santos-Neto and A. P. C. S. Costa, "Enterprise maturity models: a systematic literature review," *Enterprise Information Systems*, vol. 13, no. 5, pp. 719–769, 2019.

- [76] A. V. Looy, M. D. Backer, and G. Poels, “Defining business process maturity. A journey towards excellence,” *Total Quality Management & Business Excellence*, vol. 22, no. 11, pp. 1119–1137, 2011.
- [77] G. d. S. S. Leal, W. Guédria, and H. Panetto, “Interoperability assessment: A systematic literature review,” *Computers in Industry*, vol. 106, pp. 111–132, 2019.
- [78] L. Von Bertalanffy, “The history and status of general systems theory,” *Academy of management journal*, vol. 15, no. 4, pp. 407–426, 1972.
- [79] F. Heylighen, “Basic concepts of the systems approach,” *Principia Cybernetica*, Brussels, 1998.
- [80] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, “Capability maturity model, version 1.1,” *IEEE software*, vol. 10, no. 4, pp. 18–27, 1993.
- [81] J. K. Crawford and Others, *Project management maturity model*. Taylor & Francis, 2007.
- [82] A. De Carolis, M. Macchi, E. Negri, and S. Terzi, “A maturity model for assessing the digital readiness of manufacturing companies,” in *IFIP International Conference on Advances in Production Management Systems*, pp. 13–20, Springer, 2017.
- [83] B. K. Anggoro, M. Hubeis, and I. Sailah, “Information system interoperability maturity model,” *Bulletin of Social Informatics Theory and Application*, vol. 2, no. 1, pp. 22–33, 2018.
- [84] S. U. Team, “Standard cmmi appraisal method for process improvement (scampi) a, version 1.3: Method definition document,” *Software Engineering Institute, Carnegie Mellon University, Tech. Rep. CMU/SEI-2011-HB-001*, 2011.
- [85] B. Barafort, A. Rousseau, and E. Dubois, “How to design an innovative framework for process improvement? The TIPA for ITIL case,” in *European Conference on Software Process Improvement*, pp. 48–59, Springer, 2014.
- [86] L. Yue, P. Niu, Y. Fang, and Z. Han, “Towards a smart manufacturing maturity assessment framework: a socio-technical perspective,” in *Journal of Physics: Conference Series*, vol. 1345, p. 042063, IOP Publishing, 2019.
- [87] M. Romero, W. Guédria, H. Panetto, and B. Béatrix, “Towards Smart Assessment: A Metamodel Proposal,” in *15th OTM/IFAC/IFIP International Workshop on Enterprise Integration, Interoperability and Networking (EI2N 2019)*, pp. 12–22, Springer, oct 2019.
- [88] G. Leal, W. Guédria, and H. Panetto, “A semi-automated system for interoperability assessment: an ontology-based approach,” *Enterprise Information Systems*, vol. 14, pp. 308–333, Mar. 2020.
- [89] Alalwan and Thomas, “An Ontology-based Approach to Assessing Records Management Systems,” *e-Service Journal*, vol. 8, no. 3, p. 24, 2013.
- [90] M. Ghazanfari, M. Jafari, and S. Rouhani, “A tool to evaluate the business intelligence of enterprise systems,” *Scientia Iranica*, vol. 18, no. 6, pp. 1579–1590, 2011.

-
- [91] R. Almeida, I. Percheiro, C. Pardo, and M. M. da Silva, “An ontology-based model for itil process assessment using tipa for itil,” in *International Conference on Software Process Improvement and Capability Determination*, pp. 104–118, Springer, 2018.
- [92] G. da Silva Serapião Leal, W. Guédria, and H. Panetto, “A semi-automated system for interoperability assessment: an ontology-based approach,” *Enterprise Information Systems*, vol. 14, no. 3, pp. 308–333, 2020.
- [93] D. Proença and J. Borbinha, “Information Governance Maturity Assessment Using Enterprise Architecture Model Analysis and Description Logics,” in *International Conference on Theory and Practice of Digital Libraries*, pp. 265–279, Springer, 2019.
- [94] ISO Central Secretary, “IEC 15504-2: Information Technology-Process assessment-Part 2: Performing an Assessment,” 2004.
- [95] R. Oberhauser, “Leveraging semantic web computing for context-aware software engineering environments,” in *Semantic Web*, IntechOpen, 2010.
- [96] R. Gove and J. Uzdziński, “A performance-based system maturity assessment framework,” *Procedia Computer Science*, vol. 16, pp. 688–697, 2013.
- [97] J. Vásquez, S. Aguirre, C. E. Fuquene-Retamoso, G. Bruno, P. C. Priarone, and L. Settinieri, “A conceptual framework for the eco-efficiency assessment of small-and medium-sized enterprises,” *Journal of Cleaner Production*, vol. 237, p. 117660, 2019.
- [98] G. Žigienė, E. Rybakovas, and R. Alzbutas, “Artificial intelligence based commercial risk management framework for smes,” *Sustainability*, vol. 11, no. 16, p. 4501, 2019.
- [99] P. Salimzadeh and J. Courvisanos, “A conceptual framework for assessing sustainable development in regional SMEs,” *Journal of Environmental Assessment Policy and Management*, vol. 17, no. 04, p. 1550039, 2015.
- [100] C. Lee and B. Nowell, “A framework for assessing the performance of nonprofit organizations,” *American Journal of Evaluation*, vol. 36, no. 3, pp. 299–319, 2015.
- [101] J. vom Brocke, M.-S. Baier, T. Schmiedel, K. Stelzl, M. Röglinger, and C. Wehking, “Context-aware business process management,” *Business & Information Systems Engineering*, pp. 1–18, 2021.
- [102] M. Gruninger and M. S. Fox, “The logic of enterprise modelling,” in *Modelling and Methodologies for Enterprise Integration*, pp. 140–157, Springer, 1996.
- [103] P. Pullonen, J. Tom, R. Matulevičius, and A. Toots, “Privacy-enhanced bpmn: Enabling data privacy analysis in business processes models,” *Software and Systems Modeling*, vol. 18, no. 6, pp. 3235–3264, 2019.
- [104] D. Proença and J. Borbinha, “Using enterprise architecture model analysis and description logics for maturity assessment,” SAC ’18, (New York, NY, USA), p. 102–109, Association for Computing Machinery, 2018.
- [105] K. Ougaabal, G. Zacharewicz, Y. Ducq, and S. Tazi, “Path simulation in bpmn workflow using resource aggregation,” in *I-ESA Workshop Pathways towards a Modelling and Architecture Language for Interoperable Cyber-Physical Systems*, 2020.

- [106] A. C. Fauzan, R. Sarno, and N. F. Ariyani, "Structure-based ontology matching of business process model for fraud detection," in *2017 11th International Conference on Information & Communication Technology and System (ICTS)*, pp. 221–226, IEEE, 2017.
- [107] S. I. Besik and J.-C. Freytag, "Ontology-Based Privacy Compliance Checking for Clinical Workflows," 2019.
- [108] O. M. Kherbouche, A. Ahmad, M. Bouneffa, and H. Basson, "Ontology-based change impact assessment in dynamic business processes," in *2013 11th International Conference on Frontiers of Information Technology*, pp. 235–240, IEEE, 2013.
- [109] R. Almeida, R. Lourinho, M. M. da Silva, and R. Pereira, "A model for assessing COBIT 5 and ISO 27001 simultaneously," in *2018 IEEE 20th Conference on Business Informatics (CBI)*, vol. 1, pp. 60–69, IEEE, 2018.
- [110] A. Ligeza and T. Potempa, "Artificial intelligence for knowledge management with bpmn and rules," in *IFIP International Workshop on Artificial Intelligence for Knowledge Management*, pp. 19–37, Springer, 2012.
- [111] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, and A. G. Frank, "The expected contribution of Industry 4.0 technologies for industrial performance," *International Journal of Production Economics*, vol. 204, pp. 383–394, 2018.
- [112] S. M. Weiss, C. A. Kulikowski, S. Amarel, and A. Safir, "A model-based method for computer-aided medical decision-making," *Artificial intelligence*, vol. 11, no. 1-2, pp. 145–172, 1978.
- [113] T.-C. Chang and R. A. Wysk, *Computer-aided manufacturing*. Prentice Hall PTR, 1997.
- [114] B. Raphael and I. F. C. Smith, *Fundamentals of computer-aided engineering*. John Wiley & sons, 2003.
- [115] J. Zhou, P. Li, Y. Zhou, B. Wang, J. Zang, and L. Meng, "Toward New-Generation Intelligent Manufacturing," *Engineering*, vol. 4, no. 1, pp. 11–20, 2018.
- [116] H. Panetto, B. B. Iung, D. Ivanov, G. Weichhart, and X. Wang, "Challenges for the cyber-physical manufacturing enterprises of the future," *Annual Reviews in Control*, 2019.
- [117] R. F. da Silva, V. J. Salles, and R. O. Prates, "SMART: uma ferramenta de apoio a projetos de sistemas colaborativos com foco em impactos sociais," in *Proceedings of the IX Symposium on Human Factors in Computing Systems*, pp. 253–254, Brazilian Computer Society, 2010.
- [118] S. Dong and V. R. Kamat, "Robust mobile computing framework for visualization of simulated processes in augmented reality," in *Proceedings of the Winter Simulation Conference*, pp. 3111–3122, Winter Simulation Conference, 2010.
- [119] W. T. Crow, M. J. van den Berg, G. J. Huffman, and T. Pellarin, "Correcting rainfall using satellite-based surface soil moisture retrievals: The Soil Moisture Analysis Rainfall Tool (SMART)," *Water Resources Research*, vol. 47, no. 8, 2011.
- [120] W. Wang and Y. Chen, "SmartPeerCast: a Smart QoS driven P2P live streaming framework," *Multimedia Tools and Applications*, vol. 54, pp. 445–471, aug 2011.

-
- [121] D. Zeinalipour-Yazti, C. Laoudias, M. I. Andreou, and D. Gunopulos, "Disclosure-free GPS trace search in smartphone networks," in *2011 IEEE 12th International Conference on Mobile Data Management*, vol. 1, pp. 78–87, IEEE, 2011.
- [122] S. Koutra, V. Becue, and C. S. Ioakimidis, "Searching for the 'smart' definition through its spatial approach," *Energy*, vol. 169, pp. 924–936, 2019.
- [123] H. S. Kang, J. Y. Lee, S. Choi, H. Kim, J. H. Park, J. Y. Son, B. H. Kim, and S. Do Noh, "Smart manufacturing: Past research, present findings, and future directions," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 1, pp. 111–128, 2016.
- [124] H. Lovell and H. Lovell, "The promise of smart grids," *Local Environment*, vol. 0, no. 0, pp. 1–15, 2018.
- [125] B. Blau, J. Kramer, T. Conte, and C. Van Dintner, "Service value networks," in *2009 IEEE conference on commerce and enterprise computing*, pp. 194–201, IEEE, 2009.
- [126] Q. Li, Q. Tang, I. Chan, H. Wei, Y. Pu, H. Jiang, J. Li, and J. Zhou, "Smart manufacturing standardization: Architectures, reference models and standards framework," *Computers in Industry*, vol. 101, pp. 91–106, 2018.
- [127] M. El Hendy, S. Miniaoui, and H. Fakhry, "Towards strategic information & communication technology (ICT) framework for smart cities decision-makers," in *2015 2nd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, pp. 1–7, IEEE, 2015.
- [128] D. Wolter and A. Kirsch, "Smart Environments: What is it and Why Should We Care?," *KI-Künstliche Intelligenz*, vol. 31, no. 3, pp. 231–237, 2017.
- [129] R. Moss Kanter and S. S. Litow, "Informed and interconnected: A manifesto for smarter cities," *Harvard Business School General Management Unit Working Paper*, no. 09-141, 2009.
- [130] R. Jucevicius and G. Jucevicius, "From Knowledge to Smart City: A Conceptual Study," in *European Conference on Knowledge Management*, vol. 2, p. 508, Academic Conferences International Limited, 2014.
- [131] C. Garau, "From territory to smartphone: Smart fruition of cultural heritage for dynamic tourism development," *Planning Practice and Research*, vol. 29, no. 3, pp. 238–255, 2014.
- [132] S. Al-Nasrawi, C. Adams, and A. El-Zaart, "Smartness of Smart Sustainable Cities: A Multidimensional Dynamic Process fostering Sustainable Development," in *Proc. of the 5th International Conference on Smart Cities, Systems, Devices and Technologies (SMART 2016)*, 2016.
- [133] V. Della Corte, M. Aria, and G. Del Gaudio, "Smart, open, user innovation and competitive advantage: A model for museums and heritage sites," *Museum Management and Curatorship*, vol. 32, no. 1, pp. 50–79, 2017.
- [134] T. Nam and T. A. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions," in *Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times*, pp. 282–291, ACM, 2011.

- [135] M. Cavada, D. V. Hunt, and C. D. Rogers, “Smart cities: Contradicting definitions and unclear measures,” in *World Sustainability Forum*, pp. 1–12, 2014.
- [136] K. Douaioui, M. Fri, C. Mabroukki, and E. A. Semma, “The interaction between industry 4.0 and smart logistics: concepts and perspectives,” in *2018 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)*, vol. 0021266798, pp. 128–132, apr 2018.
- [137] D. Schäfer, D. Schilder, and D. Scha, “Smart capital in German start-ups – an empirical analysis,” vol. 1066, 2009.
- [138] F. Jaramillo, J. P. Mulki, and J. S. Boles, “Workplace Stressors, Job Attitude, and Job Behaviors: Is Interpersonal Conflict the Missing Link?,” *Journal of Personal Selling & Sales Management*, vol. 31, no. 3, pp. 339–356, 2011.
- [139] D. Kumar, “Application of modern tools and techniques for mine safety & disaster management,” *Journal of The Institution of Engineers (India): Series D*, vol. 97, no. 1, pp. 77–85, 2016.
- [140] B. Hatt, “Street Smarts vs. Book Smarts: The Figured World of Smartness in the Lives of Marginalized, Urban Youth,” *The Urban Review*, vol. 39, pp. 145–166, jun 2007.
- [141] T. Dunleavy, “High School Algebra Students Busting the Myth about Mathematical Smartness: Counterstories to the Dominant Narrative “Get It Quick and Get It Right”,” *Education Sciences*, vol. 8, no. 2, p. 58, 2018.
- [142] M. Herczeg, “The Smart, the Intelligent and the Wise: Roles and Values of Interactive Technologies,” in *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia*, IITM ’10, (New York, NY, USA), pp. 17–26, ACM, 2010.
- [143] S. Alter, “Making Sense of Smartness in the Context of Smart Devices and Smart Systems,” *Information Systems Frontiers*, pp. 1–13, 2019.
- [144] Y. Naudet, W. Guedria, and D. Chen, “Systems science for enterprise interoperability,” in *2009 International Conference on Interoperability for Enterprise Software and Applications China*, pp. 107–113, IEEE, 2009.
- [145] J. L. G. Dietz and J. A. P. Hoogervorst, “Enterprise ontology in enterprise engineering,” in *Proceedings of the 2008 ACM symposium on Applied computing*, pp. 572–579, ACM, 2008.
- [146] A. Caragliu, C. D. Bo, and P. Nijkamp, “Smart Cities in Europe,” *Journal of Urban Technology*, vol. 18, no. 2, pp. 65–82, 2011.
- [147] Y. Patel and N. Doshi, “Social implications of smart cities,” *Procedia Computer Science*, vol. 155, pp. 692–697, 2019.
- [148] R. E. Hall, B. Bowerman, J. Braverman, J. Taylor, H. Todosow, and U. Von Wimmersperg, “The vision of a smart city,” tech. rep., Brookhaven National Lab., Upton, NY (US), 2000.
- [149] C. Lim and P. P. Maglio, “Data-driven understanding of smart service systems through text mining,” *Service Science*, vol. 10, no. 2, pp. 154–180, 2018.

-
- [150] Li Jiang, Da-You Liu, and Bo Yang, "Smart home research," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, vol. 2, pp. 659–663 vol.2, aug 2004.
- [151] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *Journal of Network and Computer Applications*, vol. 97, pp. 48–65, 2017.
- [152] S. Davidoff, M. K. Lee, C. Yiu, J. Zimmerman, and A. K. Dey, "Principles of Smart Home Control," in *UbiComp 2006: Ubiquitous Computing* (P. Dourish and A. Friday, eds.), (Berlin, Heidelberg), pp. 19–34, Springer Berlin Heidelberg, 2006.
- [153] V. P. Vyshnevs'kyi and S. I. Knyazev, "Smart industry: prospects and challenges," *Economy of Ukraine*, vol. 7, pp. 22–37, 2017.
- [154] N. Kaur and S. K. Sood, "Cognitive decision making in smart industry," *Computers in Industry*, vol. 74, pp. 151–161, 2015.
- [155] Y. Liao, F. Deschamps, E. d. F. R. Loures, and L. F. P. Ramos, "Past, present and future of Industry 4.0—a systematic literature review and research agenda proposal," *International journal of production research*, vol. 55, no. 12, pp. 3609–3629, 2017.
- [156] P. P. Maglio, S. L. Vargo, N. Caswell, and J. Spohrer, "The service system is the basic abstraction of service science," *Information Systems and e-business Management*, vol. 7, no. 4, pp. 395–406, 2009.
- [157] P. P. Maglio and C.-H. Lim, "Innovation and big data in smart service systems," *Journal of Innovation Management*, vol. 4, no. 1, pp. 11–21, 2016.
- [158] A. Medina-Borja, "Smart Things as Service Providers: A Call for Convergence of Disciplines to Build a Research Agenda for the Service Systems of the Future," *Service Science*, vol. 7, no. 1, pp. ii–v, 2015.
- [159] P. P. Maglio, S. K. Kwan, and J. Spohrer, "Commentary—Toward a research agenda for human-centered service system innovation," *Service Science*, vol. 7, no. 1, pp. 1–10, 2015.
- [160] A. Arroub, B. Zahi, E. Sabir, and M. Sadik, "A literature review on Smart Cities: Paradigms, opportunities and open problems," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 180–186, IEEE, 2016.
- [161] A. Saad al sumaiti, M. H. Ahmed, and M. M. A. Salama, "Smart home activities: A literature review," *Electric Power Components and Systems*, vol. 42, no. 3-4, pp. 294–305, 2014.
- [162] V. S. Gunge and P. S. Yalagi, "Smart home automation: A literature review," *International Journal of Computer Applications*, vol. 975, p. 8887, 2016.
- [163] L. Liu, E. Stroulia, I. Nikolaidis, A. Miguel-Cruz, and A. R. Rincon, "Smart homes and home health monitoring technologies for older adults: A systematic review," *International journal of medical informatics*, vol. 91, pp. 44–59, 2016.
- [164] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, 2017.

- [165] D. Beverungen, O. Müller, M. Matzner, J. Mendling, and J. vom Brocke, “Conceptualizing smart service systems,” *Electronic Markets*, vol. 29, no. 1, pp. 7–18, 2019.
- [166] OMG, “Business Process Model and Notation (BPMN), Version 2.0,” jan 2011.
- [167] R. Armstrong, B. J. Hall, J. Doyle, and E. Waters, “‘Scoping the scope’ of a cochrane review,” *Journal of Public Health*, vol. 33, no. 1, pp. 147–150, 2011.
- [168] P. Robinson and J. Lowe, “Literature reviews vs systematic reviews,” *Australian and New Zealand journal of public health*, vol. 39, no. 2, p. 103, 2015.
- [169] B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, and S. Linkman, “Systematic literature reviews in software engineering – A tertiary study,” *Information and Software Technology*, vol. 52, no. 8, pp. 792–805, 2010.
- [170] S. S. Kamble, A. Gunasekaran, and S. A. Gawankar, “Sustainable Industry 4.0 framework: A systematic literature review identifying the current trends and future perspectives,” *Process Safety and Environmental Protection*, vol. 117, pp. 408–425, 2018.
- [171] W. Kraaij and R. Pohlmann, “Viewing stemming as recall enhancement,” in *SIGIR*, vol. 96, pp. 40–48, 1996.
- [172] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.
- [173] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [174] H. Steinhaus, “Sur la division des corp materiels en parties,” *Bull. Acad. Polon. Sci*, vol. 1, no. 804, p. 801, 1956.
- [175] G. H. Ball and D. J. Hall, “ISODATA, a novel method of data analysis and pattern classification,” tech. rep., Stanford research inst Menlo Park CA, 1965.
- [176] J. MacQueen and Others, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.
- [177] S. Lloyd, “Least squares quantization in PCM,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [178] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [179] S. Haykin, *Neural networks*, vol. 2. Prentice hall New York, 1994.
- [180] T. Bures, D. Weyns, B. Schmerl, E. Tovar, E. Boden, T. Gabor, I. Gerostathopoulos, P. Gupta, and E. Kang, “Software Engineering for Smart Cyber-Physical Systems : Challenges and Promising Solutions,” vol. 42, no. 2, pp. 19–24, 2017.
- [181] P. T. Hoff, “‘Fool me once, shame on you; fool me twice, shame on me’: African American students’ reclamation of smartness as resistance,” *Race Ethnicity and Education*, vol. 19, no. 6, pp. 1200–1208, 2016.

-
- [182] J. Aguilar, M. Jerez, M. Mendonca, and M. Sánchez, “MiSCi: autonomic reflective middleware for smart cities,” in *International Conference on Technologies and Innovation*, pp. 241–253, Springer, 2016.
- [183] R. Wlezien, G. Horner, A. McGowan, S. Padula, M. Scott, R. Silcox, and J. Simpson, “The aircraft morphing program,” in *39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, p. 1927, 1998.
- [184] M. D. Smith and M. E. Paté-Cornell, “Cyber risk analysis for a smart grid: how smart is smart enough? A multiarmed bandit approach to cyber security investment,” *IEEE Transactions on Engineering Management*, vol. 65, no. 3, pp. 434–447, 2018.
- [185] S. Kadambe and Q. Jiang, “Environment Aware Chemical Detection & classification,” *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, pp. 55–59 Vol.1, 2004.
- [186] E. Brusa and A. Calà, “Identifying the smartness of a mechatronic coiler through the ‘ Systems Engineering ’,” 2014.
- [187] W. Barfield and K. Baird, “Issues in the design and use of wearable computers,” *Virtual Reality*, vol. 3, no. 3, pp. 157–166, 1998.
- [188] V. Kumar, A. Dixit, R. R. G. Javalgi, and M. Dass, “Research framework, strategies, and applications of intelligent agent technologies (IATs) in marketing,” *Journal of the Academy of Marketing Science*, vol. 44, pp. 24–45, jan 2016.
- [189] H. Sujan, B. A. Weitz, and N. Kumar, “Learning Orientation, Working Smart, and Effective Selling,” *Journal of Marketing*, vol. 58, no. 3, pp. 39–52, 1994.
- [190] K. Martin, “‘We did all the work’: seeing smartness in a Poarch Creek way,” *Race Ethnicity and Education*, vol. 19, no. 6, pp. 1177–1185, 2016.
- [191] A. Chang, “Resisting the Orthodox Smart Label: High School Latinas and the Redefinition of Smartness on the Western Frontier,” *Journal of Latinos and Education*, vol. 16, no. 1, pp. 30–40, 2017.
- [192] S. J. Rosenholtz and C. Simpson, “The Formation of Ability Conceptions: Developmental Trend or Social Construction?,” *Review of Educational Research*, vol. 54, no. 1, pp. 31–63, 1984.
- [193] V. Albino, U. Berardi, and R. M. Dangelico, “Smart cities: Definitions, dimensions, performance, and initiatives,” *Journal of urban technology*, vol. 22, no. 1, pp. 3–21, 2015.
- [194] J. Hvorecký, “University Leadership and Role Distribution,” in *15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pp. 1–8, 2017.
- [195] P. Jansson, “Smartness as prudence : smart power and classical realism,” *Journal of Political Power*, vol. 11, no. 3, pp. 341–358, 2018.
- [196] P. De Vettor, M. Mrissa, D. Benslimane, and S. Berbar, “A Service Oriented Architecture for Linked Data Integration,” in *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, pp. 198–203, IEEE, 2014.

- [197] J. Li, J. Jin, D. Yuan, M. Palaniswami, and K. Moessner, “EHOPES: Data-centered Fog platform for smart living,” in *2015 International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 308–313, IEEE, 2015.
- [198] N. Mlalila, D. M. Kadam, H. Swai, and A. Hilonga, “Transformation of food packaging from passive to innovative via nanotechnology: concepts and critiques,” *Journal of Food Science and Technology*, vol. 53, pp. 3395–3407, sep 2016.
- [199] P. Lison and A. Kutuzov, “Redefining context windows for word embedding models: An experimental study,” *arXiv preprint arXiv:1704.05781*, 2017.
- [200] H. Caselles-Dupré, F. Lesaint, and J. Royo-Letelier, “Word2vec applied to recommendation: Hyperparameters matter,” in *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 352–356, ACM, 2018.
- [201] D. Anguita, “Smart adaptive systems: State of the art and future directions of research,” in *Proceedings of the European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems-EUNITE*, pp. 1–4, 2001.
- [202] D. Tokody, “Digitising the European industry - holonic systems approach,” *Procedia Manufacturing*, vol. 22, pp. 1015–1022, 2018.
- [203] G.-J. Cheng, L.-T. Liu, X.-J. Qiang, and Y. Liu, “Industry 4.0 development and application of intelligent manufacturing,” in *2016 international conference on information system and artificial intelligence (ISAI)*, pp. 407–410, IEEE, 2016.
- [204] F. Longo, L. Nicoletti, and A. Padovano, “Ubiquitous knowledge empowers the Smart Factory: The impacts of a Service-oriented Digital Twin on enterprises’ performance,” *Annual Reviews in Control*, 2019.
- [205] A. Abuarqoub, H. Abusaimah, M. Hammoudeh, D. Uliyan, M. A. Abu-Hashem, S. Murad, M. Al-Jarrah, and F. Al-Fayez, “A survey on internet of things enabled smart campus applications,” in *Proceedings of the International Conference on Future Networks and Distributed Systems*, p. 50, ACM, 2017.
- [206] Y. Geum, H. Jeon, and H. Lee, “Developing new smart services using integrated morphological analysis: integration of the market-pull and technology-push approach,” *Service Business*, vol. 10, no. 3, pp. 531–555, 2016.
- [207] L. Huang and L. Huang, “Intelligence of Smart Systems: Model, Bounds, and Algorithms,” *IEEE/ACM Trans. Netw.*, vol. 25, pp. 2960–2973, oct 2017.
- [208] L. F. Z. Rivera, M. M. Larrondo-Petrie, and L. R. Da Silva, “Implementation of cloud-based smart adaptive remote laboratories for education,” in *2017 IEEE Frontiers in Education Conference (FIE)*, pp. 1–5, IEEE, 2017.
- [209] J. Rasmussen, “A Framework for Cognitive Task Analysis in Systems Design,” in *Intelligent Decision Support in Process Environments* (E. Hollnagel, G. Mancini, and D. D. Woods, eds.), (Berlin, Heidelberg), pp. 175–196, Springer Berlin Heidelberg, 1986.
- [210] L. Deshayes, A. Vacher, L. Welsch, and A. Donmez, “Towards an information processing representation for dynamic process optimization in smart machining systems,” *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 1, pp. 219–227, nov 2007.

-
- [211] J. A. Slotwinski and R. B. Tilove, “Smart Assembly: Industry Needs and Challenges,” in *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*, PerMIS '07, (New York, NY, USA), pp. 257–262, ACM, 2007.
- [212] B. Denkena, H. Henning, and L.-E. Lorenzen, “Genetics and intelligence: new approaches in production engineering,” *Production Engineering*, vol. 4, pp. 65–73, feb 2010.
- [213] W. Jakob, J. Á. G. Ordiano, N. Ludwig, R. Mikut, and V. Hagenmeyer, “Towards Coding Strategies for Forecasting-based Scheduling in Smart Grids and the Energy Lab 2.0,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, (New York, NY, USA), pp. 1271–1278, ACM, 2017.
- [214] D. Romero and O. Noran, “Towards Green Sensing Virtual Enterprises: Interconnected Sensing Enterprises, Intelligent Assets and Smart Products in the Cyber-Physical Circular Economy,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11719–11724, 2017.
- [215] A. T. Jones, D. Romero, and T. Wuest, “Modeling agents as joint cognitive systems in smart manufacturing systems,” *Manufacturing Letters*, vol. 17, pp. 6–8, 2018.
- [216] M. N. Yuniarto and A. W. Labib, “Designing an online and real time simulation, control and monitoring of disturbances in an intelligent manufacturing system,” in *IEEE International Conference on Industrial Informatics, 2003. INDIN 2003. Proceedings.*, pp. 273–278, IEEE, 2003.
- [217] G. Schreiber, B. Wielinga, and J. Breuker, *KADS: A principled approach to knowledge-based system development*, vol. 11. Academic Press, 1993.
- [218] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [219] V. López-Morales and O. López-Ortega, “A distributed semantic network model for a collaborative intelligent system,” *Journal of Intelligent Manufacturing*, vol. 16, no. 4-5, pp. 515–525, 2005.
- [220] S. Bagaveyev and D. J. Cook, “Designing and evaluating active learning methods for activity recognition,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pp. 469–478, ACM, 2014.
- [221] P. Brödner, ““Super-intelligent” machine: technological exuberance or the road to subjection,” *AI & SOCIETY*, vol. 33, no. 3, pp. 335–346, 2018.
- [222] D. Weihrauch, P. A. Schindler, and W. Sihm, “A Conceptual Model for Developing a Smart Process Control System,” in *Procedia CIRP*, vol. 67, (Fraunhofer Austria Research GmbH, Theresianumgasse 7, Wien, 1040, Austria), pp. 386–391, 2018.
- [223] S. Corley, M. Tesselaar, J. Cooley, J. Meinköhn, F. Malabocchia, and F. Garijo, “The application of intelligent and mobile agents to network and service management,” in *International Conference on Intelligence in Services and Networks*, pp. 127–138, Springer, 1998.
- [224] Z. Wang, L. Jiao, P. Yan, X. Wang, J. Yi, and X. Shi, “Research and development of intelligent cutting database cloud platform system,” *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9-12, pp. 3131–3143, 2018.

- [225] Z. Irani and M. M. Kamal, "Intelligent Systems Research in the Construction Industry," *Expert Systems with Applications*, vol. 41, no. 4, Part 1, pp. 934–950, 2014.
- [226] G. Weichhart, A. Molina, D. Chen, L. E. Whitman, and F. Vernadat, "Challenges and current developments for Sensing, Smart and Sustainable Enterprise Systems," *Computers in Industry*, vol. 79, pp. 34–46, 2016.
- [227] J. Miranda, P. Ponce, A. Molina, and P. Wright, "Sensing, smart and sustainable technologies for Agri-Food 4.0," *Computers in Industry*, vol. 108, pp. 21–36, 2019.
- [228] W. R. Ashby, *Principles of the Self-Organizing System*, pp. 521–536. Boston, MA: Springer US, 1991.
- [229] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," in *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, WMCSA '94, (Washington, DC, USA), pp. 85–90, IEEE Computer Society, 1994.
- [230] A. K. Dey, G. D. Abowd, and A. Wood, "CyberDesk: A framework for providing self-integrating context-aware services," *Knowledge-based systems*, vol. 11, no. 1, pp. 3–13, 1998.
- [231] P. Y.-z. Wan, "Emergence à la Systems Theory: Epistemological Totalausschluss or Ontological Novelty?," *Philosophy of the Social Sciences*, vol. 41, no. 2, pp. 178–210, 2011.
- [232] P. Corning, "The re-emergence of "emergence": A venerable concept in search of a theory," *Complexity*, vol. 7, pp. 18–30, 2002.
- [233] C. W. Johnson, "What are emergent properties and how do they affect the engineering of complex systems?," *Reliability Engineering and System Safety*, vol. 91, no. 12, pp. 1475–1481, 2006.
- [234] J. Rumbaugh, I. Jacobson, and G. Booch, *Unified modeling language reference manual, the*. Pearson Higher Education, 2004.
- [235] M. Romero, W. Guédria, H. Panetto, and B. Barafort, "Towards a Conceptual Framework for Smart Assessment in Organisations," in *Proceedings of the IFAC World Conference 2020*, IFAC, 2020.
- [236] J. M. Gascueña, E. Navarro, and A. Fernández-Caballero, "Model-driven engineering techniques for the development of multi-agent systems," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 1, pp. 159–173, 2012.
- [237] V. H. Publishing, *Archimate(r) 3.1 Specification*. The Open Group series, Van Haren Publishing, 2020.
- [238] B. E. Board, "The Guide to the Systems Engineering Body of Knowledge (SEBoK)," 2017.
- [239] L. Von Bertalanffy, "General system theory," *New York*, vol. 41973, no. 1968, p. 40, 1968.
- [240] D. Chen, "Enterprise Interoperability Framework.," in *EMOI-INTEROP*, 2006.
- [241] ISO Central Secretary, "ISO 14258: Industrial automation systems - Concepts and rules for enterprise models," standard, International Organization for Standardization, Geneva, CH, mar 1998.

-
- [242] Y. Naudet, T. Latour, W. Guedria, and D. Chen, “Towards a systemic formalisation of interoperability,” *Computers in Industry*, vol. 61, no. 2, pp. 176–185, 2010.
- [243] C. Peters, G. Castellano, M. Rehm, E. André, A. Raouzaïou, K. Rapantzikos, K. Karpouzis, G. Volpe, A. Camurri, and A. Vasalou, “Fundamentals of agent perception and attention modelling,” in *Emotion-Oriented Systems*, pp. 293–319, Springer, 2011.
- [244] D. Chavarría-Barrientos, L. M. Camarinha-Matos, and A. Molina, “Achieving the sensing, smart and sustainable “everything,”” in *Working Conference on Virtual Enterprises*, pp. 575–588, Springer, 2017.
- [245] M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017.
- [246] F. Baader, D. McGuinness, D. Nardi, and P. Patel-Schneider, *The Description Logics Handbook*. Cambridge University Press, 2003.
- [247] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, “Large scale incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.
- [248] Y. Wang, B. Wang, Y. Hou, and D. Song, “Automatic update of ontology concept hierarchy with new entity insertion and new concept generation based on semantic measurement,” in *Proceedings of the 2018 7th International Conference on Software and Computer Applications, ICSCA 2018*, (New York, NY, USA), p. 274–278, Association for Computing Machinery, 2018.
- [249] A. P. Sheth, “Changing focus on interoperability in information systems: from system, syntax, structure to semantics,” in *Interoperating geographic information systems*, pp. 5–29, Springer, 1999.
- [250] ISO Central Secretary, “ISO/IEC 15504-2: Information Technology - Process assessment - Part 2: Performing an Assessment,” standard, International Organization for Standardization, 2004.
- [251] ISO Central Secretary, “ISO 9001: Quality management systems - Requirements,” standard, International Organization for Standardization, Geneva, CH, sep 2015.
- [252] ISO Central Secretary, “ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering,” *ISO/IEC/IEEE 29148:2018(E)*, pp. 1–104, 2018.
- [253] M. D. Mesarovic, “General systems theory and its mathematical foundation,” tech. rep., CASE WESTERN RESERVE UNIV CLEVELAND OHSYSTEMS RESEARCH CENTER, 1967.
- [254] A. Tarhan and G. Giray, “On the Use of Ontologies in Software Process Assessment: A Systematic Literature Review,” in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, EASE’17*, (New York, NY, USA), pp. 2–11, ACM, 2017.

- [255] C. P. Team, “Capability maturity model®integration (CMMI SM), version 1.1,” *CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1. 1)*, 2002.
- [256] C. Feilmayr and W. Wöß, “An analysis of ontologies and their success factors for application to business,” *Data & Knowledge Engineering*, vol. 101, pp. 1–23, 2016.
- [257] L. Ehrlinger and W. Wöß, “Towards a Definition of Knowledge Graphs.,” *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48, 2016.
- [258] D. Proença and J. Borbinha, “Formalizing ISO/IEC 15504-5 and SEI CMMI v1. 3—Enabling automatic inference of maturity and capability levels,” *Computer Standards & Interfaces*, vol. 60, pp. 13–25, 2018.
- [259] C. C. Aggarwal and C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.
- [260] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [261] M. H. Hassoun and Others, *Fundamentals of artificial neural networks*. MIT press, 1995.
- [262] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, “Supervised machine learning: A review of classification techniques,” *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.
- [263] S. Klatt and B. Bohnet, “You Don’t Have to Think Twice if You Carefully Tokenize,” in *Natural Language Processing – IJCNLP 2004* (K.-Y. Su, J. Tsujii, J.-H. Lee, and O. Y. Kwong, eds.), (Berlin, Heidelberg), pp. 299–309, Springer Berlin Heidelberg, 2005.
- [264] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.
- [265] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [266] A. Y. Ng, “Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance,” in *Proceedings of the Twenty-First International Conference on Machine Learning, ICML ’04*, (New York, NY, USA), p. 78, Association for Computing Machinery, 2004.
- [267] P. M. Institute, “A guide to the project management body of knowledge (PMBOK guide),” tech. rep., Project Management Institute, Newtown Square, Pa: Project Management Institute, 2004.
- [268] G. Antoniou and F. Van Harmelen, “Web ontology language: Owl,” in *Handbook on ontologies*, pp. 67–92, Springer, 2004.
- [269] D. Brickley, R. V. Guha, and B. McBride, “Rdf schema 1.1. W3C recommendation,” *World Wide Web Consortium, February*, 2014.
- [270] F. Manola, E. Miller, B. McBride, and Others, “RDF primer,” *W3C recommendation*, vol. 10, no. 1-107, p. 6, 2004.

-
- [271] B. Smith, “Ontology,” in *The furniture of the world*, pp. 47–68, Brill Rodopi, 2012.
- [272] M. A. Musen, “The protégé project: a look back and a look forward,” *AI matters*, vol. 1, no. 4, pp. 4–12, 2015.
- [273] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, M. Dean, and Others, “SWRL: A semantic web rule language combining OWL and RuleML,” *W3C Member submission*, vol. 21, no. 79, pp. 1–31, 2004.
- [274] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [275] D. M. Hawkins, “The problem of overfitting,” *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.
- [276] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A Practical OWL-DL Reasoner,” *Web Semant.*, vol. 5, pp. 51–53, jun 2007.
- [277] A. Fedosejev, *React.js essentials*. Packt Publishing Ltd, 2015.
- [278] F. Peres, J. Yang, and M. Ghazel, “A formal framework for the formalization of informal requirements,” *The International Journal of Soft Computing and Software Engineering*, vol. 2, no. 8, pp. 14–27, 2012.
- [279] V. R. Basili, “Applying the Goal/Question/Metric paradigm in the experience factory,” *Software Quality Assurance and Measurement: A Worldwide Perspective*, vol. 7, no. 4, pp. 21–44, 1993.
- [280] C. Ghidini, M. Rospocher, and L. Serafini, “A formalisation of BPMN in description logics,” *FBK-irst, Tech. Rep. TR*, pp. 4–6, 2008.
- [281] ISO Central Secretary, “ISO/IEC 27001: Information Security Management,” standard, International Organization for Standardization, Geneva, CH, mar 2013.
- [282] J. Samalikova, R. J. Kusters, J. J. M. Trienekens, and A. J. M. M. Weijters, “Process mining support for Capability Maturity Model Integration-based software process assessment, in principle and in practice,” *Journal of Software: Evolution and Process*, vol. 26, no. 7, pp. 714–728, 2014.
- [283] W. Van Der Aalst, *Process mining: discovery, conformance and enhancement of business processes*, vol. 2. Springer, 2011.
- [284] A. Awadid, D. Bork, and S. Nurcan, “Towards Assessing the Multi-view Modeling Capability of Enterprise Modeling Methods,” in *IFIP Working Conference on The Practice of Enterprise Modeling*, pp. 351–361, Springer, 2018.
- [285] C. Kleissner, “Enterprise Objects Framework: a second generation object-relational enabler,” in *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pp. 455–459, 1995.
- [286] G. S. S. Leal, W. Guédria, and H. Panetto, “Enterprise interoperability assessment: a requirements engineering approach,” *International Journal of Computer Integrated Manufacturing*, pp. 1–22, 2020.

- [287] R. Lalitha, B. Latha, and G. Sumathi, “A Methodology to Establish a Conceptual Model and to Prioritise the Activities during Software Analysis,” *Asian Journal of Research in Social Sciences and Humanities*, vol. 6, no. 10, pp. 2114–2127, 2016.
- [288] E. A. Emerson and J. Y. Halpern, ““Sometimes” and “not never” revisited: on branching versus linear time temporal logic,” *Journal of the ACM (JACM)*, vol. 33, no. 1, pp. 151–178, 1986.
- [289] E. M. Clarke and E. A. Emerson, “Design and synthesis of synchronization skeletons using branching time temporal logic,” in *Workshop on Logic of Programs*, pp. 52–71, Springer, 1981.
- [290] J. L. Peterson, “Petri nets,” *ACM Computing Surveys (CSUR)*, vol. 9, no. 3, pp. 223–252, 1977.
- [291] G. Li, R. M. de Carvalho, and W. M. van der Aalst, “Automatic discovery of object-centric behavioral constraint models,” in *International conference on business information systems*, pp. 43–58, Springer, 2017.
- [292] B. F. van Dongen, R. M. Dijkman, and J. Mendling, *Measuring similarity between business process models*. Technische Universiteit Eindhoven, 2007.
- [293] B. Van Dongen, R. Dijkman, and J. Mendling, “Measuring similarity between business process models,” in *Seminal Contributions to Information Systems Engineering*, pp. 405–419, Springer, 2013.
- [294] W.-N. Lee, N. Shah, K. Sundlass, and M. Musen, “Comparison of ontology-based semantic-similarity measures,” in *AMIA annual symposium proceedings*, vol. 2008, p. 384, American Medical Informatics Association, 2008.
- [295] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, “Ontology matching: A machine learning approach,” in *Handbook on ontologies*, pp. 385–403, Springer, 2004.
- [296] M. A. Khoudja, M. Fareh, and H. Bouarfa, “Ontology matching using neural networks: survey and analysis,” in *2018 International Conference on Applied Smart Systems (ICASS)*, pp. 1–6, IEEE, 2018.
- [297] D. Reißner, R. Conforti, M. Dumas, M. La Rosa, and A. Armas-Cervantes, “Scalable conformance checking of business processes,” in *On the Move to Meaningful Internet Systems. OTM 2017 Conferences*, (Cham), pp. 607–627, Springer International Publishing, 2017.
- [298] M. ELLEUCH, N. ASSY, N. LAGA, W. GAALOUL, O. ALAOU I SMAILI, and B. Benatallah, “A meta model for mining processes from email data,” in *2020 IEEE International Conference on Services Computing (SCC)*, pp. 152–161, 2020.
- [299] M. Elleuch, O. A. Ismaili, N. Laga, W. Gaaloul, and B. Benatallah, “Discovering activities from emails based on pattern discovery approach,” in *Business Process Management Forum* (D. Fahland, C. Ghidini, J. Becker, and M. Dumas, eds.), (Cham), pp. 88–104, Springer International Publishing, 2020.
- [300] P. Hake, M. Zapp, P. Fettke, and P. Loos, “Supporting business process modeling using rnns for label classification,” in *International Conference on Applications of Natural Language to Information Systems*, pp. 283–286, Springer, 2017.

-
- [301] G. A. Miller, “Wordnet: a lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [302] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” *IEEE transactions on neural networks and learning systems*, 2020.
- [303] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” *arXiv preprint arXiv:1509.09292*, 2015.
- [304] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [305] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [306] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *arXiv preprint arXiv:1706.03762*, 2017.
- [307] P. Lara, M. Sánchez, and J. Villalobos, “Enterprise modeling and operational technologies (ot) application in the oil and gas industry,” *Journal of Industrial Information Integration*, vol. 19, p. 100160, 2020.
- [308] D. Karagiannis and H. Kühn, “Metamodelling platforms,” in *EC-Web*, vol. 2455, p. 182, 2002.
- [309] M. Brambilla, J. Cabot, and M. Wimmer, “Model-driven software engineering in practice,” *Synthesis lectures on software engineering*, vol. 3, no. 1, pp. 1–207, 2017.
- [310] T. Clark, P. Sammut, and J. Willans, “Applied metamodelling: a foundation for language driven development.,” 2008.
- [311] H. Panetto and J.-F. Pétrin, “Metamodelling of production systems process models using uml stereotypes,” *International Journal of Internet and Enterprise Management*, vol. 3, no. 2, pp. 155–169, 2005.
- [312] A. Giovannini, A. Aubry, H. Panetto, H. El Haouzi, L. Pierrel, and M. Dassisti, “Anti-logicist framework for design-knowledge representation,” *Annual Reviews in Control*, vol. 39, pp. 144–157, 2015.
- [313] M. Dassisti, A. Giovannini, P. Merla, M. Chimienti, and H. Panetto, “An approach to support industry 4.0 adoption in smes using a core-metamodel,” *Annual Reviews in control*, vol. 47, pp. 266–274, 2019.
- [314] B. A. Yilma, Y. Naudet, and H. Panetto, “A new paradigm and meta-model for cyber-physical-social systems,” in *21st IFAC World Congress, IFAC 2020* (IFAC, ed.), vol. 53 of *IFAC Papers Online, 21st IFAC World Congress, IFAC 2020 Proceedings, IFAC-PapersOnLine*, (Berlin, Germany), pp. 10949–10954, Elsevier, July 2020.
- [315] P. Bernus, L. Nemes, and R. Morris, “The meaning of an enterprise model,” in *Modelling and methodologies for Enterprise Integration*, pp. 183–200, Springer, 1996.

- [316] E. Trunzer, A. Wullenweber, and B. Vogel-Heuser, “Graphical modeling notation for data collection and analysis architectures in cyber-physical systems of systems,” *Journal of Industrial Information Integration*, vol. 19, p. 100155, 2020.
- [317] ISO, “19439, 2006-enterprise integration-framework for enterprise modelling,” *International Standards Organization, Geneva, Switzerland*, 2006.
- [318] P. Bernus, “Some thoughts on enterprise modelling,” *Production Planning & Control*, vol. 12, no. 2, pp. 110–118, 2001.
- [319] S. H. Lim, N. Juster, and A. de Pennington, “Enterprise modelling and integration: a taxonomy of seven key aspects,” *Computers in Industry*, vol. 34, no. 3, pp. 339–359, 1997.
- [320] F. Vernadat, “Enterprise modeling in the context of enterprise engineering: State of the art and outlook,” *International Journal of Production Management and Engineering*, vol. 2, no. 2, pp. 57–73, 2014.
- [321] C. Menzel and R. J. Mayer, “The idef family of languages,” in *Handbook on architectures of information systems*, pp. 209–241, Springer, 1998.
- [322] A.-W. Scheer, “Architecture of integrated information systems (aris),” in *Business Process Engineering*, pp. 4–16, Springer, 1994.
- [323] G. Keller, A.-W. Scheer, and M. Nüttgens, *Semantische Prozeßmodellierung auf der Grundlage " Ereignisgesteuerter Prozeßketten (EPK) "*. Inst. für Wirtschaftsinformatik, 1992.
- [324] F. Vernadat, “UEML: towards a unified enterprise modelling language,” *International Journal of Production Research*, vol. 40, no. 17, pp. 4309–4321, 2002.
- [325] V. Anaya, G. Berio, M. Harzallah, P. Heymans, R. Matulevicius, A. Opdahl, H. Panetto, M. J. Verdecho, and B. Vallespir, “DEM4: Tutorial on UEML,” 2007.
- [326] G. Doumeingts, “méthode de conception des systèmes en productique,” *Automatic Control, Université de Bordeaux I*, 1984.
- [327] D. Chen and G. Doumeingts, “The grai-gim reference model, architecture and methodology,” in *Architectures for Enterprise Integration*, pp. 102–126, Springer, 1996.
- [328] E. C. AMICE, “Management overview,” *CIMOSA: Open System Architecture for CIM*, pp. 9–29, 1993.
- [329] F. Vernadat, “The cimoso languages,” in *Handbook on architectures of information systems*, pp. 243–263, Springer, 1998.
- [330] F. Vernadat, “Cimoso: Enterprise modelling and enterprise integration using a process-based approach,” in *Proceedings of the JSPE/IFIP TC5/WG5. 3 Workshop on the Design of Information Infrastructure Systems for Manufacturing*, pp. 65–84, 1993.
- [331] B. Vallespir and Y. Ducq, “Enterprise modelling: from early languages to models transformation,” *International Journal of Production Research*, vol. 56, no. 8, pp. 2878–2896, 2018.
- [332] T. O. Group, “Business Process Model And Notation 2.0 Specification,” *The Open Group*, 2011.

-
- [333] S. A. White, “Introduction to BPMN,” *Ibm Cooperation*, vol. 2, no. 0, p. 0, 2004.
- [334] A. Josey, M. Lankhorst, I. Band, H. Jonkers, and D. Quartel, “An introduction to the archimate® 3.0 specification,” *White Paper from The Open Group*, 2016.
- [335] L. Graesser and W. L. Keng, *Foundations of deep reinforcement learning: theory and practice in Python*. Addison-Wesley Professional, 2019.
- [336] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [337] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, pp. 2672–2680, 2014.
- [338] C. L. Giles, G. M. Kuhn, and R. J. Williams, “Dynamic recurrent neural networks: Theory and applications,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 153–156, 1994.
- [339] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, and Others, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies,” 2001.
- [340] F. Almeida and G. Xexéo, “Word embeddings: A survey,” *arXiv preprint arXiv:1901.09069*, 2019.
- [341] Y. Bengio, J.-S. Senécal, and Others, “Quick Training of Probabilistic Neural Nets by Importance Sampling,” in *AISTATS*, pp. 1–9, 2003.
- [342] M. Baroni, G. Dinu, and G. Kruszewski, “Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 238–247, 2014.
- [343] G. Salton, A. Wong, and C. S. Yang, “A Vector Space Model for Automatic Indexing,” *Commun. ACM*, vol. 18, pp. 613–620, nov 1975.
- [344] T. K. Landauer, P. W. Foltz, and D. Laham, “An introduction to latent semantic analysis,” *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.
- [345] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [346] B. A. Yilma, N. Aghenda, M. Romero, Y. Naudet, and H. Panetto, “Personalised visual art recommendation by learning latent semantic representations,” in *2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA)*, pp. 1–6, 2020.
- [347] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.
- [348] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

- [349] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
- [350] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.
- [351] A. Sperduti and A. Starita, “Supervised neural networks for the classification of structures,” *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 714–735, 1997.
- [352] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020.
- [353] R. Akerkar and P. Sajja, *Knowledge-based systems*. Jones & Bartlett Publishers, 2009.
- [354] A. Gangemi and V. Presutti, “Ontology design patterns,” in *Handbook on ontologies*, pp. 221–243, Springer, 2009.
- [355] M. C. A. Klein and D. Fensel, “Ontology versioning on the Semantic Web.,” in *SWWS*, pp. 75–91, 2001.
- [356] A. Maedche and S. Staab, “Ontology learning for the semantic web,” *IEEE Intelligent systems*, vol. 16, no. 2, pp. 72–79, 2001.
- [357] N. Masri, Y. A. Sultan, A. N. Akkila, A. Almasri, A. Ahmed, A. Y. Mahmoud, I. Zaqout, and S. S. Abu-Naser, “Survey of Rule-Based Systems,” *International Journal of Academic Information Systems Research (IJAIRS)*, vol. 3, no. 7, pp. 1–23, 2019.
- [358] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, “HermiT: an OWL 2 reasoner,” *Journal of Automated Reasoning*, vol. 53, no. 3, pp. 245–269, 2014.

Appendices

A

Enterprise Modelling

An enterprise model is a structured representation of the state of an enterprise considering one or more organisational aspects such as processes, structure, information, resources, goals, constraints and government activity [102] as a means of understanding and analysing the enterprise [307]. The objective of enterprise models is to provide organisations the capability to perform enterprise design, analysis and operation with models as foundation, following a model-driven approach [102]. In this sense, a model is an abstraction of one or more aspects of a real-world system, situation or phenomenon [47]. Hence, it serves as a approximation of the reality at a given point of time. The purpose of a model is to reflect relevant features and characteristics of a system that is the objective of [47]. In the context of assessments in organisations, enterprise models are one of the possible data sources that can be collected and analysed. Note that the information given by an enterprise model is highly dependent on the modelling language that was used to design it. Therefore, understanding the characteristics and the differences between different enterprise modelling languages is relevant if models are going to be used as source of information during an appraisal. This appendix describes concepts related to modelling languages, enterprise modelling, and two specific enterprise modelling languages that are used in this thesis: BPMN and ArchiMate.

A.1 Modelling Languages

A modelling language is a language composed of a set of elements, with which a model can be described, that express information regarding the structure of a system or its behaviour [308]. The elements of a modelling language are represented through a Syntax that provides a description of the modelling method and the rules to combine the modelling elements, the Semantics, which describe the meaning of those elements, and the Notations, which define their visual representation [308]. Every element of the syntax of a modelling language requires the definition of its semantics to represent its meaning and a notation defining its visual representation [308].

The elements of a modelling language are graphically represented in Figure A.1. Note that the language is considered as a component of a Modelling Technique, which itself is a component of a Modelling Method that also defines Mechanisms and Algorithms used for modelling, which provide the functionality to use and evaluate the models [308]. Another component of the Modelling Technique is the Modelling Procedure, which defines the steps that must be performed for applying the language to generate models.

Other information sources, for instance [309] refer to the Notation as the Concrete Syntax, which describes specific representations of the language, considering visual elements that can be

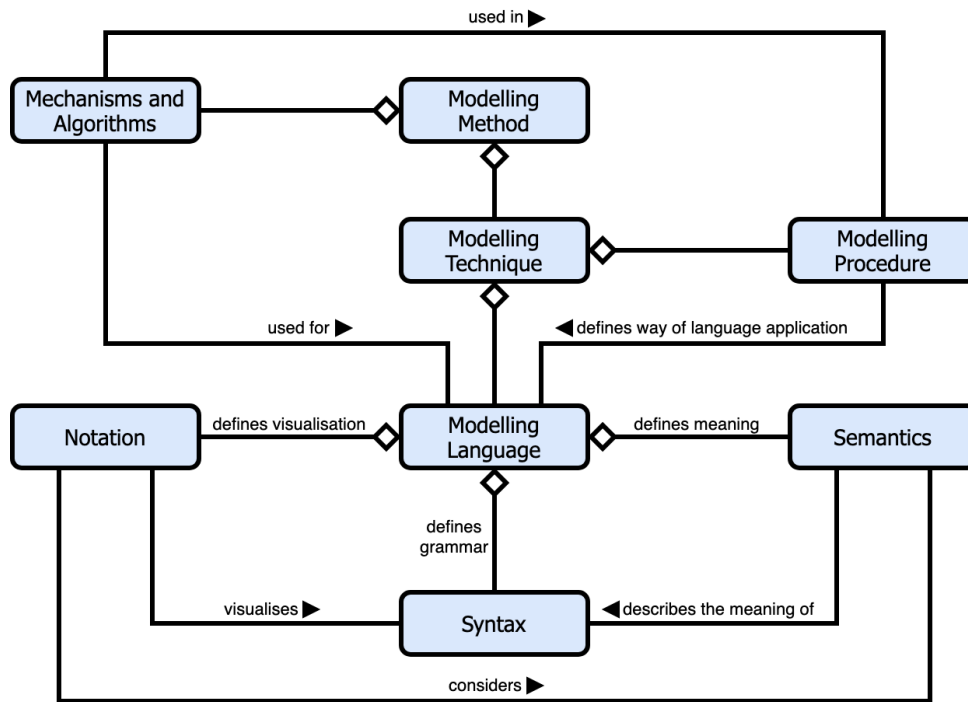


Figure A.1: Elements of a modelling language and other related aspects (adapted from [308]).

graphical or textual. The Syntax as shown in Figure A.1 is therefore considered as the Abstract syntax, describing the structure of the language and how the primitives are combined. It is worth mentioning that there could be more than one Concrete Syntax for a language, thus there may exist various textual or graphical representations for the same language, depending on the specific needs of the possible users.

Models described through modelling languages could be used for sketching, which means that they serve only for communication purposes; for blueprinting, which implies that they are used to define the specifications of a certain system; and for programming, where the models are able to be automatically transformed into code in a software development context [309]. Models can be static, presenting details addressing the structure of a system in terms of entities and relationships between them; or dynamic, describing behavioural aspects of the entities of the system including their interactions [309].

In some cases, a modelling language may not provide all the required features or the expression capability to represent specific scenarios or systems, this leads to the development of extended versions of such languages or the design of languages that are specifically focused on supporting the design of instances of those systems [309]. In this context, modelling languages can be grouped into the following categories:

- **Domain-Specific Modelling Languages (DSMLs).** Designed to model a specific domain, context, or enterprise, with the objective to aid people that must model things in that domain.
- **General-Purpose Modelling Languages (GPMLs).** Its modelling notations can be used to represent any area or domain for modelling purposes.

A relevant aspect in the context of modelling languages is meta-modelling. When using a modelling language, not only regular models can be built, but also meta-models, which are

models that represent other models. For instance, a model is an abstract representation of the real world, and a meta-model acts as an abstraction of that model. This meta-model allows to define the properties and features for representing the model. Indeed, a model must conform to its meta-model analogously to a natural sentence, which must be compliant to the rules defined by a certain grammar. This also occurs to modelling languages, in which there exist meta-models used for their definition [309]. This conceptualisation of meta-models is particularly present in the software engineering domain, in which meta-modelling is often related to the act of defining modelling languages [310]. Likewise, meta-modelling languages in the model-driven software engineering context have been proposed throughout the years such as the Meta Object Facility (MOF) ¹² and the Ecore from the Eclipse Modelling Framework (EMF) ¹³.

On the other hand, another perspective of meta-modelling is focused on viewing meta-models as conceptual models that define concepts, relationships and semantics that allow to create more concrete models [87, 311–314]. In this sense, the meta-model and the model are models with different levels of abstraction, the former representing a system from a certain domain, and the later describing a particular instance of that system. In this thesis we follow this trend, thus we introduce meta-models from a conceptual modelling perspective.

A.2 Enterprise Modelling Languages

Modelling languages are tightly connected to enterprise models since they provide the elements required to describe them. Indeed, enterprise modelling uses multiple languages, methods and tools to support the enterprise engineering process, which is a strategic large-scale design effort that allows to enterprises to act, in an integrated and coherent manner, towards the achievement of the business goals [315].

Enterprise Models are expressed using semi-formal Enterprise modelling languages (EMLs) [47] or with simple graphical notations. Different types of models can be obtained through the use of EMLs such as resource models, organisational models, structural diagrams, control sequence diagrams, among others [47]. Enterprise models can be processed by computers and they can have static or dynamic nature, depending on the purpose of the model [47]. Moreover, they have proven to be useful in several areas that comprise the design, engineering, operations, monitoring or maintenance activities of enterprise systems, and data collection and analysis architectures [47, 316].

In enterprise modelling, languages and frameworks for modelling must rely on general principles such as model accuracy and relevance, and more specific ones described as follows according to [47], which were summarised from diverse sources [317–320]:

- **Plural nature of enterprise models.** Given the complexity of an enterprise, several models are necessary to represent its aspects. Hence, the enterprise model is the combination of these granular models.
- **Principle of minimalist ontology.** All modelling languages must be defined through a minimal and non-redundant set of modelling elements, with each one composed of a minimal set of attributes and properties.
- **Concept of modelling views.** Efficient modelling methods should provide the minimal amount of non-overlapping views (which can be considered as mechanisms that allow to

¹²<https://www.omg.org/spec/MOF/2.5.1/About-MOF/>

¹³<https://wiki.eclipse.org/Ecore>

focus on specific aspects of the system that are the objective of analysis, excluding others to overcome the complexity of the system) to all relevant aspects of the studied entity.

- **Concept of modelling levels.** They represent levels of abstraction devised for a specific population of users given by the details of the system to be taken into consideration. They can be viewed as phases or steps of the life cycle of the modelling project.
- **Separation of enterprise behaviour and enterprise functionality.** Enterprise behaviour reflects the way the enterprise performs its work, whilst the enterprise functionality describes the tasks or activities that are executed to perform the work. These two concepts should be separated so as to foster enterprise agility.
- **Late binding of resources to process steps.** Resources must not be linked to process steps, and specially not to activities, in an *a priori* manner, instead they should be assigned to activities in run-time.
- **Fundamental types of flows.** Enterprise modelling languages and tools should differentiate the material/physical, information/decision, and control flows that exist in an enterprise. In this sense, two other flows can be considered, the financial flow and the energy flow, the last one dealing with aspects such as the use of electricity, gas, etc.

There exists several modelling languages and methods used in the context of enterprises. Unified Modelling Language (UML) [234], for instance, comprises a family of languages that were originally intended to specify, visualise, and document models of software systems but it is widely used for modelling non-software systems, including business models. Another family of languages is IDEF (Integrated Computer-Aided Manufacturing DEFinition), which comprises a family of modelling languages addressing systems and software engineering [321]. These languages cover several aspects such functional modelling (IDEF0), information modelling (IDEF1), dynamic modelling (IDEF2), data modelling (IDEF1X), among others. Examples of languages that are more specific for the business context are Business Process Model and Notation (BPMN) [166], which provides a graphical representation of business processes in a business process model; and ArchiMate [237], which provides means to describe, analyse and visualise the architecture of an enterprise through a broad view based on separating enterprise aspects into layers. Likewise, ARIS (Architecture of Integrated Information Systems) [322] also provides methods to perform analysis of business processes from a holistic perspective. It is based on a five-view architecture known as ARIS house, which is composed of the following views: organisation, data, product/service, function, and process. The idea behind this segmentation is to reduce the complexity of the model with the objective to simplify the modelling process. ARIS is strongly associated to the Event-driven Process Chains (EPC) modelling language [323], which was developed within the framework of ARIS. It allows to represent events, functions and other modelling elements such as process owner, logical connectors and relationships, etc., with the purpose of describing business processes. In this context, the large amount of modelling languages that emerged throughout the years fostered the development of a hub able to integrate different modelling languages, thus the Unified Enterprise Modelling Language (UEML) was proposed [324] with the objective to serve as an interface between enterprise modelling languages, tools, and frameworks [325]. On the other hand, considering a Computer-integrated Manufacturing (CIM) context, the GRAI method [326] was devised to design models describing decision systems. GRAI models are composed of a macro reference model for manufacturing systems, which is decomposed into a decision sub-system, an information sub-system and a physical sub-system; and a micro reference model, which allows to represent internal elements of a decision centre. Originally,

GRAI was developed with a focus on the modelling of Production Management Systems, but it was extended during the following years to consider the entire Manufacturing System. This resulted in GIM (GRAI Integrated Methodology), which is intended to support the design of Integrated Manufacturing Systems with the objective to deduce the specifications of such type of systems [327]. Another modelling approach focused on enterprise integration is CIMOSA [328], which structures a CIM system as a set of processes that intercommunicate and are executed by a set of functional entities [329]. CIMOSA proposes a paradigm based on an event-driven process-based modelling approach with the objective to include requirements definition, system design and implementation description [330]. The CIMOSA languages allow to represent different modelling levels considering business processes at the core of the modelling approach. It includes workflow, functional, information, resource, and organisation languages. In the context of assessments in organisations, enterprise models are one of the data sources that can be collected and processed to provide the assessment results. Among other factors, the information provided by a model is highly dependant on the modelling language used to design it. For instance, BPMN is strongly focused on describing process execution aspects, whilst ArchiMate provides a more global view of the relationships between architecture domains through a layer-based structure. For more information regarding the contributions and relevant research addressing enterprise modelling of the last four decades the reader may refer to the works by [331] and [47].

The experiments and case studies presented throughout this thesis are specifically focused on two modelling languages: BPMN and ArchiMate. The following sections present a brief description of each of them.

A.2.1 Business Process Model and Notation (BPMN)

BPM is a standard for business process modelling. Its primary goal is to provide a notation that could be understandable by all business users comprising business analysts, technical developers responsible for implementing the technology that will perform or provide assistance to perform processes, and business people responsible for managing and monitoring processes. It also has the objective to ensure that XML languages designed to execute business process can be properly visualised [332]. BPMN defines a Business Process Diagram (BPD) based on a flowcharting technique that allows to create graphical models describing business processes [333].

The BPM specification is structured in layers in which there is a *Core* layer including the most fundamental elements of the modelling language, these elements are required for constructing BPDs [332]. The Core layer contains three sub-packages: (1) **Foundation**, including the fundamental constructs needed for BPMN, (2) **Service**, including the fundamental constructs needed for modelling services and interfaces, and (3) **Common**, including classes that are common to the layers of Process, Choreography, and Collaboration. The three packages mentioned before include the core elements to define BPDs, we briefly describe each of them next.

- **Process**. It describes a sequence of *Activities* in an organisation. It is depicted as a graph of *Flow Elements* that could be *Activities*, *Events*, *Gateways* and *Sequence Flows*. Processes can be represented at any level of the organisation, from enterprise-wide Process to Processes performed by a single person.
- **Choreography**. It is a type of process but it is different from a standard Process in terms of purpose and behaviour. It is devised for formalising the manner business *Participants* coordinate their interactions, its objective is not to orchestrate the work performed by the *Participants* but to reflect the exchange of information (*Messages*) between them.

- **Collaboration.** It contains classes useful for modelling Collaborations, which is a set of Participants shown as *Pools* and their interactions, represented through *Message Flows*. A Collaboration could also include *Processes* within the *Pools* and/or *Choreographies* between the *Pools*.

Each layer contains its own set of modelling elements that allow to define BPDs for specific needs. In this work, we focus on the Process package since the method validation presented treats elements within this package. However, we argue that the proposed approach could also be suitable for the treatment of BPDs with elements from the Choreography and Collaboration packages as well. For a more detailed description of the elements included by each layer we encourage the reader to refer to the original BPMN specification [332].

A.2.2 ArchiMate

ArchiMate is a modelling language that provides a set of default elements for describing, analysing and communicating diverse aspects of Enterprise Architecture. The design of the language has started from a set of generic concepts, which have after been specialised towards application at different architectural layers [237]. Figure A.2 presents the organisation of the layers within the scope of the ArchiMate framework [334].

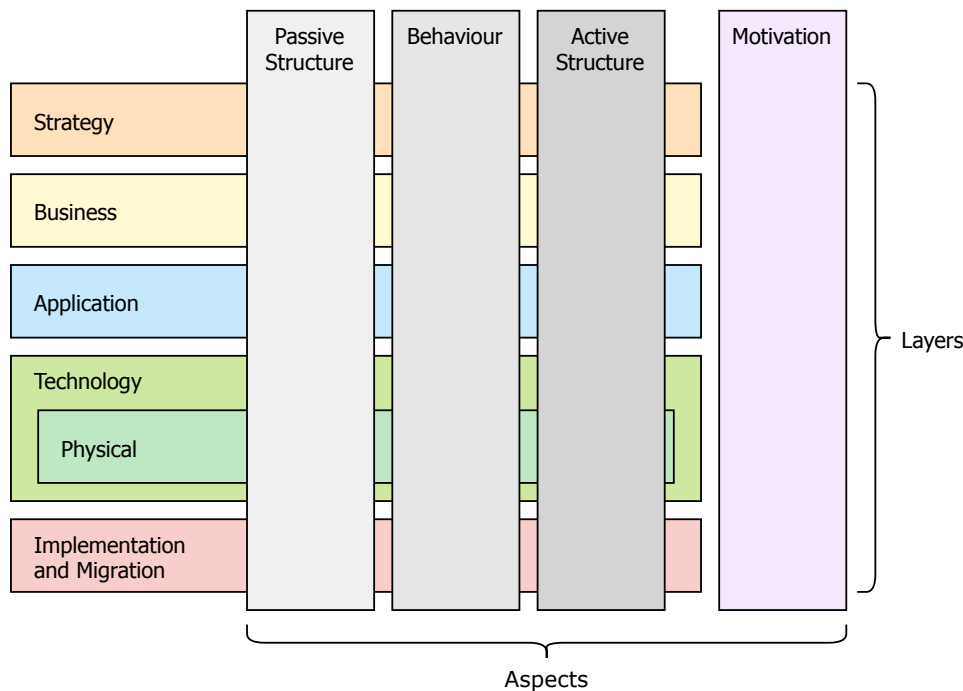


Figure A.2: Framework of ArchiMate (adapted from [237]).

The layered-structure of the framework is intended to model different viewpoints of the enterprise. The core layers of ArchiMate are:

- **Business Layer.** Describes business services offered to customers, which are realised by business processes performed by business actors.
- **Application Layer.** Describes application services supporting the business, and the applications that allow to realise those services.

- **Technology Layer.** Describes technology services needed to run the applications, and the computer and communication hardware and system software that realise those services. Moreover, physical elements are also included, and they can be considered as a sub-layer from the Technology Layer.

In addition of the three core layers, the full ArchiMate framework adds two more layers:

- **Strategy Layer.** Provides elements to model strategic directions and choices.
- **Implementation and Migration Layer.** Includes elements for modelling migration planning, and implementation programs and projects to support program, portfolio, and project management.

Moreover, as part of the full framework ArchiMate also allows to represent the Motivation elements that drive the design and operation of the enterprise. They describe the way the Enterprise Architecture is aligned to its context. The stakeholder, value, meaning, driver, assessment, and goal elements are some examples of Motivation elements.

The framework also includes the following aspects that are shown in columns in Figure A.2:

- **Active Structure.** Represents structural elements such as business actors, application components, devices that display behaviour, etc.
- **Behaviour.** Represents behaviour of the actors in the form of processes, functions, events, and services. Structural elements are assigned to behavioural elements.
- **Passive Structure.** Represents the objects that on which a behaviour is applied. In general, objects from the passive structure aspect are information objects or data objects from the Business and Application layers, respectively, but they can also represent physical objects.

Note that ArchiMate also supports Composite Elements, which are elements that do not fit into a single aspect but instead they are a combination of two or more aspects. On the other hand, it is worth mentioning that ArchiMate does not require the modellers to use specific structures or layouts such as the one presented in this section. The framework acts as a mean to categorise and organise the available elements of the language. For an extended description of ArchiMate, the reader may refer to the ArchiMate specification 3.1¹⁴.

¹⁴<https://pubs.opengroup.org/architecture/archimate3-doc/toc.html>

B

Deep Learning

Deep Learning (DL) is a sub-field of Machine Learning [218] encompassing algorithms and techniques that are capable learning data representations with multiple levels of abstraction. This is achieved through a combination of non-linear modules to transform a low level representation into a more abstract one [13]. Typically, DL models are neural networks that are composed of multiple processing layers that stack simple modules that compute non-linear input-output mappings. These modules are subject to learning. This structure allows them to achieve a high level of generalisation capability. This is accomplished through the implementation of complex functions applied to their inputs that provides them the capability to be sensitive to the relevant details of the input data and insensitive to the irrelevant ones [13]. Figure B.1 presents a generic pipeline that is applied when using a DL model.

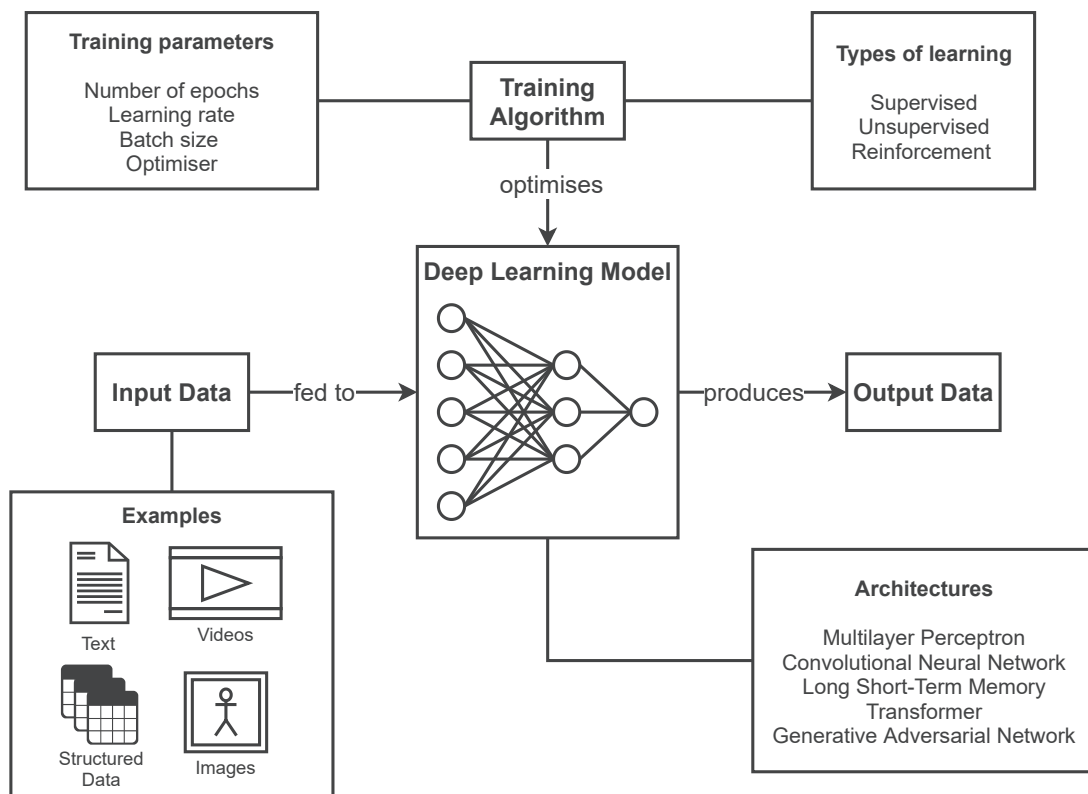


Figure B.1: Typical pipeline of a Deep Learning model.

DL models are neural networks that receive input data that may be composed of text, multimedia, structured data, etc., as shown in Figure B.1 and they produce an output result. Neural networks learn functions that map inputs to outputs. They apply a set of computations that are performed sequentially on an input to provide the output, this process is known as forward pass [335]. Neural networks learn through some training algorithm that allow to optimise their parameters. This is done by using a sufficiently representative training data that is shown to the network during training with the objective to minimise a loss function, which is selected according to the task that the network will be trained to perform. There exist three types of learning: supervised, unsupervised, and reinforcement learning. Some common training parameters that must be defined to train a DL model are the number of epochs, learning rate, and batch size. Likewise, several DL architectures have been proposed by the research community during the recent years, allowing to achieve state-of-art performances in tasks such as speech recognition, video and image classification and segmentation, object detection, language translation, etc. Some examples of these architectures are: Convolutional Neural Networks [336], Long Short-Term Memory Networks [260], Transformers [306], and Generative Adversarial Networks [337]. For more details addressing the foundations of DL, the reader could refer to [13]. The rest of this appendix presents brief descriptions of the three main DL-related approaches that are used in this work: Long Short-term Memory, Word Embedding, and Graph Neural Network.

B.1 Long Short-term Memory (LSTM)

An LSTM is a special type of Recurrent Neural Network (RNN) [338] that is capable of learning long-term dependencies through a complex structure compared to typical standard RNNs. This structure is based on hidden units known as memory cells [260]. The internal structure of the memory cells allows them to overcome the vanishing (or exploding) gradient problem [339], allowing the network to retain dependencies from very early inputs. That is, it is able to effectively treat very long sequences of input data. An LSTM memory cell is shown in Figure B.2.

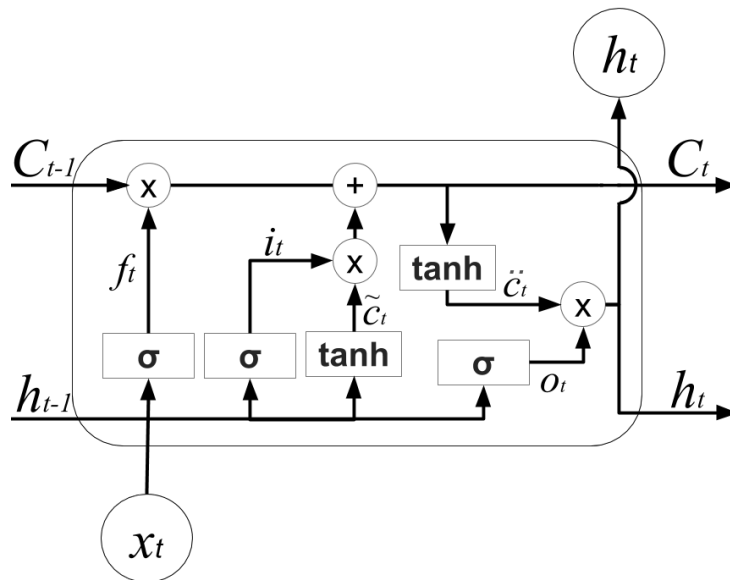


Figure B.2: An LSTM memory cell.

LSTM cells allow to optionally remove or save information. This is accomplished through

a mechanism based on the use of gates. An LSTM cell contains three gates: one to forget information, another for updating the previous cell state, and the last one to generate the output of the current cell state. Gates are composed of a Sigmoid neural network layer and a point-wise product operation. Equations B.1 to B.6 present the mechanism of an LSTM.

$$\vec{f}_t = \sigma(\vec{W}_f[\vec{h}_{t-1}, \vec{x}_t] + \vec{b}_f), \quad (\text{B.1})$$

$$\vec{i}_t = \sigma(\vec{W}_i[\vec{h}_{t-1}, \vec{x}_t] + \vec{b}_i), \quad (\text{B.2})$$

$$\tilde{\vec{C}}_t = \tanh(\vec{W}_c[\vec{h}_{t-1}, \vec{x}_t] + \vec{b}_c), \quad (\text{B.3})$$

$$\vec{C}_t = \vec{f}_t \circ \vec{C}_{t-1} + \vec{i}_t \circ \tilde{\vec{C}}_t, \quad (\text{B.4})$$

$$\vec{o}_t = \sigma(\vec{W}_o[\vec{h}_{t-1}, \vec{x}_t] + \vec{b}_o), \quad (\text{B.5})$$

$$\vec{h}_t = \vec{o}_t \circ \tanh(\vec{C}_t), \quad (\text{B.6})$$

Where \vec{f}_t , \vec{i}_t , \vec{o}_t and \vec{C}_t are the forget gate, input gate (also known as update gate), output gate, and the internal cell state, respectively. The weights matrices are \vec{W}_f , \vec{W}_i , \vec{W}_c and \vec{W}_o , whilst the bias terms are defined by \vec{b}_f , \vec{b}_i , \vec{b}_c and \vec{b}_o . The activation functions Sigmoid and hyperbolic tangent are represented by σ and \tanh , respectively. The symbol \circ represents the Hadamard product operation.

LSTM networks are widely used for solving problems such as time series prediction, speech recognition, handwriting recognition, video classification, etc. In this thesis, an LSTM network is mainly used to treat text data. In this case, a text corpus is treated by the network considering that it is composed of a sequence of words or characters.

B.2 Word Embedding

In Natural Language Processing (NLP) [50], a common task is to represent words (or set of words considered as documents) using numerical values. This representation allows to apply operations such as addition, subtraction, or distance measures, whilst also making them available to be used in the context of Machine Learning algorithms [340]. The models that allow to obtain vector representation of words are known as vector space models.

There are two categories of methods for generating word vectors, depending on the strategies used to define the model that generates the vectors [341, 342]. The *count-based* methods use global information of documents such as word counts and frequencies [340]. There are several methods to generate vectors within this category such as the Term frequency-inverse document frequency weights [343], Latent Semantic Analysis (LSA) [344], the Latent Dirichlet Allocation (LDA) [345, 346], etc.

On the other hand, the *prediction-based* models encompass methods that leverage local data by training language models to generate *word embeddings*. The idea behind word embeddings is to train a model, usually a deep neural network, to perform a certain task, and use the output of one of the internal layers of the network to represent a word (generally the first layer). Hence, the embedding of a word is the projection of its raw word vector into a layer of a neural network [340]. This layer is usually named embedding layer. The first work that introduced this kind of approach was the one presented by [347], in which a neural probabilistic language model was described. Since then, similar approaches to generate word embedding have been

introduced in the literature such as the Continuous Bag-of-Words [172] and the Skip-Gram [348], both are known as versions of the word2vec algorithm. The Skip-Gram architecture served as inspiration for the development of fastText [349], which is based on the original Skip-Gram but with word vectors that are enriched with sub-word information [349]. Another well-known model is GloVe [264], which is a word2vec-based algorithm that is centred on the use of word counts.

B.3 Graph Neural Network (GNN)

Several relationships between data can be represented through the use of graphs. A graph is a data structure composed of objects (nodes) and their relationships (edges). The most simple graph structure is based on a set of sequential node connected through edges. However, there exist more complex structures such as trees, cyclic graphs, and acyclic graphs [350]. Traditional Machine Learning applications treat graph-structured data by adding a pre-processing phase to the pipeline in which the data is transformed into a simpler representation such as vectors [350]. Notwithstanding, a lot of relevant information contained in the graph is lost after applying such transformation. Graph Neural Networks (GNNs) were proposed in [350] to solve problems that involve the use of graphs as input data. GNNs are an evolution of the Recursive Neural Networks [351], which were mainly focused on dealing with directed acyclic graphs. The main idea behind a GNN is to learn mechanisms to represent graphs, nodes, edges, and sub-graphs using low-dimensional vectors [352] that are learnable from a data-driven perspective. According to [352], the general pipeline of the use of a GNN approach to solve a problem is composed of the following steps:

1. **Find graph structure.** The graph structure of the application must be found. In some cases, the graph structure is explicitly defined in the application (molecules, knowledge graphs, physical systems, etc.), whilst in other cases the graph is implicit and it must be first defined through some mechanism.
2. **Specify graph type and scale.** Graphs can be classified in several ways; directed or undirected, homogeneous or heterogeneous, and static or dynamic. The type of the graph must be identified in order to design the proper methodology to treat the graph.
3. **Design loss function.** The task type and the training setting must be considered in this stage to define the loss function to be optimised during training. The possible tasks are: node-level, edge-level, and graph-level; they make reference to the level in which the training will be focused to perform the task (node classification, node prediction, edge classification, graph regression, graph matching). As for the training setting, the possible settings are: supervised, semi-supervised (based on giving a small amount of labelled data and a large amount of unlabelled data for training), and unsupervised learning.
4. **Build model.** The model is finally built using computational modules. In [352], three common computational modules are identified: *propagation module*, to propagate information between nodes; *sampling module* to aid during propagation for large graphs; and *pooling module* to extract information from nodes to obtain the representations of high-level sub-graphs or graphs.

C

Knowledge-based Systems

A Knowledge-based System (KBS) is a system that has been designed and implemented to act as a human expert in a given domain [353]. Moreover, a KBS is computational system that uses existing knowledge to solve problems, and it is capable of generating new data, information, or knowledge [353]. A KBS is part of the classical AI, also known as model-driven or symbolic AI, and it has two main components: a Knowledge Base (KB) and an Inference Engine (IE). The former is a repository of knowledge in various forms, whilst the later is a search program that infers the knowledge from the KB [353]. Among the various types of KB, the use of Ontologies have risen during the last years due to attempts of the industry and academia to standardise the knowledge that is contained in the web so as to improve the way that knowledge is described and shared among users. The following sections briefly describe Ontologies and Inference Engines.

C.1 Ontologies

In philosophy, ontology is the study of being, of the kinds and structures of objects, properties, events and relations [271]. In the context of information science, ontologies are artefacts that describe some world for some purpose [354]. They allow to describe entities with attributes and relationships through some formal expression [271]. Ontologies can be considered as KBs that are building blocks for the Semantic Web, since they represent shareable and reusable knowledge describing a specific domain [355]. In this sense, they serve as metadata schemas, providing a vocabulary of concepts with explicitly defined semantics that can be processed by computer programs [356]. This characteristic allows them to aid people and machines to communicate effectively [356]. There are several languages devised to represent ontologies; in the context of the Semantic Web, the Web Ontology Language (OWL) is used to define ontologies using formal semantics [268]. OWL is part of the Semantic Web Stack, which is an architecture defining the set of technologies defined by the World Wide Web Consortium (W3C) for the Semantic Web.

C.2 Inference Engines

Since ontologies are often defined through machine-readable languages, they are suitable for inference. This implies that new information can be deduced from existing knowledge that was previously defined using ontologies. For this purpose, inference engines are used. An inference engine is a component of a system that uses a control structure (rule interpreter) to apply logical rules to a KB so as to infer new knowledge [357]. For OWL ontologies, a language named Semantic Web Rule Language (SWRL) [273] is used to define rules that can be applied to an

OWL ontology. SWRL allows the definition of a high-level abstract syntax for Horn-like rules in order to be combined with an OWL knowledge base. These rules have the form of an implication between an antecedent and a consequent, with a body-head structure. In this sense, each rule can be read as follows: if the conditions that are specified in the antecedent hold, then the conditions that are defined in the consequent must also hold. Once the SWRL rules have been defined for an OWL ontology, it is possible to use inference engines such as Pellet [276] or HermiT [358] to apply those rules to the ontology so as to obtain new facts and relationships.

Vers une évaluation intelligente des processus d'affaire pour l'aide à la décision en entreprises

Le monde évolue vers une nouvelle ère basée sur l'utilisation des énergies renouvelables et des technologies intelligentes [1]. Cette troisième révolution industrielle est un voyage vers l'exploitation étendue de sources d'énergie propres et les capacités des technologies intelligentes [1]. D'un point de vue technologique, cette révolution repose principalement sur la transformation numérique [2], qui met l'accent sur l'intégration de la technologie dans tous les domaines des organisations publiques et privées afin d'accroître la productivité et la création de valeur [3]. Ce paradigme est centré sur l'utilisation d'éléments omniprésents tels que les systèmes cyber-physiques [4, 5], les dispositifs de l'Internet des objets (IoT) [6, 7], etc. qui permettent de générer, consommer et traiter des données, afin d'agir sur un certain environnement ou de fournir les informations nécessaires à la prise de décision [8].

Les entreprises sont au centre de cette nouvelle ère industrielle révolutionnaire, car elles doivent s'adapter aux nouveaux défis découlant de ce nouveau scénario afin de gérer des aspects tels que la durabilité, la conformité réglementaire, l'adoption efficace de nouvelles technologies, l'optimisation des coûts, la gestion efficace des risques, etc. Ces adaptations nécessitent des transformations efficaces et, dans certains cas, considérables, qui requièrent une analyse de leurs capacités et de leurs limites comme première étape avant de planifier toute initiative de transformation. À cette fin, il est essentiel de procéder à une évaluation de l'entreprise. Si elle est effectuée correctement, elle permet d'analyser les forces et les faiblesses des capacités de l'entreprise [9], ce qui permet aux décideurs de comprendre l'état actuel de l'entreprise et, par conséquent, de planifier un état futur ciblé. Les évaluations peuvent être qualitatives ou quantitatives. Les évaluations qualitatives se concentrent sur les aspects non mesurables et sur des mesures plus subjectives [10], tandis que les évaluations quantitatives mettent l'accent sur les attributs mesurables d'un point de vue plus objectif, et sont généralement centrées sur la mesure des performances [11].

Le nouveau paradigme centré sur la technologie introduit dans le contexte de la révolution industrielle modifie non seulement les aspects de l'entreprise concernant les opérations avec les clients, les fournisseurs et les partenaires, mais aussi ceux liés aux aspects internes tels que la chaîne de production, le processus de communication, la collaboration entre les équipes, la définition de nouvelles stratégies, etc. En outre, ce scénario favorise la génération de grandes quantités de données hétérogènes, notamment des textes, des images, des vidéos, des sons, des données brutes de capteurs, des modèles d'entreprise, etc. Ces données peuvent être utilisées comme preuves pour effectuer différents types d'évaluations. Cependant, la grande variété de types de données exige de nouvelles méthodes d'évaluation et des systèmes capables de les traiter pour obtenir des résultats d'évaluation.

Dans ce contexte, en plus de fournir des moyens nouveaux, rapides et efficaces pour générer des données permettant de décrire les éléments internes et externes d'une entreprise, les technologies perturbatrices offrent également des alternatives innovantes pour effectuer des évaluations, en utilisant les données disponibles pour obtenir des vues fiables de l'état actuel des entités de l'entreprise qui peuvent être utilisées à des fins de prise de décision. Dans ce sens, le domaine de l'intelligence artificielle (IA) [12] comprend un ensemble particulièrement pertinent de techniques qui ont gagné en popularité au cours des dernières années en raison de leur capacité à obtenir des résultats de pointe pour différentes tâches. Dans un contexte industriel, les méthodes d'IA permettent de soutenir efficacement la prise de décision et, si nécessaire, d'agir dans des scénarios de plus en plus complexes. Les approches de l'IA peuvent être envisagées sous deux angles : l'IA guidée par les données et l'IA guidée par les modèles. Dans l'IA guidée par les données, les connaissances du système sont automatiquement déduites des données historiques. Elle comprend un sous-domaine appelé apprentissage profond (Deep Learning, DL) [13], qui est aujourd'hui largement répandu dans

l'industrie et le monde universitaire. Le DL est une forme d'apprentissage automatique (ML) qui permet aux ordinateurs d'apprendre à partir d'expériences passées et de comprendre certains aspects du monde par le biais d'une représentation hiérarchique des concepts [14]. Les techniques DL se caractérisent par le besoin de données pour entraîner des modèles afin de réaliser des tâches spécifiques telles que la reconnaissance vocale, la segmentation d'images, la classification de vidéos, etc. D'autre part, une perspective plus classique de l'IA comprend des techniques qui sont symboliques et fortement orientées vers l'utilisation de règles et d'heuristiques prédéfinies pour effectuer certaines tâches. La logique floue [15] et les systèmes experts [12] sont deux exemples d'approches classiques de l'IA. Dans cette thèse, ces méthodes sont considérées comme faisant partie de la catégorie de l'IA dirigée par les modèles, puisque les connaissances doivent être modélisées manuellement par des experts humains.

Ces nouvelles méthodes et technologies de traitement des données pourraient fournir des moyens de manipuler et de traiter automatiquement les preuves d'évaluation, c'est-à-dire tout type de données reflétant l'état d'une entité évaluée [16]. Le processus d'évaluation se déroule généralement selon les étapes suivantes : planification, collecte des données, détermination des résultats et présentation des résultats [17]. En particulier, la réalisation des phases de collecte de données et de détermination des résultats d'une évaluation exige souvent un investissement en termes de temps et d'efforts si elles sont effectuées par une équipe d'évaluateurs humains. L'étape de collecte de données se réfère à toutes les tâches liées à la collecte de preuves qui permettent de décrire l'entité évaluée, tandis que l'étape de détermination des résultats comprend les activités d'analyse des preuves pour fournir une vue de l'état actuel. Les initiatives d'évaluation purement manuelles sont souvent exigeantes en termes de ressources et sont plus susceptibles de produire des résultats d'évaluation incomplets, plus subjectifs, voire biaisés, étant donné que certaines preuves peuvent être collectées, traitées ou analysées de manière incorrecte par la personne chargée d'effectuer ces tâches. L'utilisation d'un système capable d'effectuer une évaluation entièrement automatisée présente trois avantages principaux : le premier point est que la collecte, le prétraitement et le filtrage des données pertinentes ne doivent pas nécessairement être effectués par des humains, ce qui entraîne une réduction significative de la consommation de ressources pour effectuer l'évaluation. De plus, si l'évaluateur humain n'a besoin que d'une preuve prétraitée, le système d'évaluation pourrait arrêter le processus à cette étape, laissant à l'expert la liberté de prendre ses propres décisions en utilisant cette preuve préalablement traitée. Le deuxième point est que le système pourrait fournir des résultats d'évaluation finaux qui pourraient être utilisés directement par les parties prenantes, ou comme une seconde opinion alternative à considérer par un décideur humain. Cette dernière option pourrait être particulièrement utile lors de la réalisation d'évaluations qualitatives. Enfin, le troisième point est que les évaluations automatisées réduisent la probabilité d'obtenir des résultats d'évaluation biaisés, incorrects, incomplets ou trop subjectifs.

Au fil des ans, plusieurs travaux ont visé à améliorer le processus d'évaluation en exploitant des technologies et des méthodologies qui représentaient une nouveauté ou une perturbation à un moment donné [18-35]. Les chercheurs et les praticiens de l'industrie ont proposé et testé différentes approches pour résoudre le problème de la réduction de l'effort humain lors de l'exécution des évaluations (qualitatives et quantitatives), tout en assurant la fiabilité et l'efficacité. Ces approches peuvent être regroupées en trois grandes catégories : les évaluations manuelles, semi-automatiques et contextuelles.

Les évaluations manuelles reposent entièrement sur la participation d'experts humains pour planifier l'évaluation, rassembler et analyser les preuves, et définir et communiquer les résultats. Les efforts de la recherche et de l'industrie concernant les évaluations manuelles se concentrent souvent sur la proposition de cadres [18-26] pour faciliter l'étude des preuves de l'évaluation ou sur la fourniture de

mécanismes formels pour planifier et réaliser les évaluations. Indépendamment de la méthodologie ou du cadre spécifique utilisé pour effectuer une évaluation, le processus consiste en une série d'étapes communes à toutes les évaluations : planification de l'évaluation, collecte des données, détermination des résultats et communication des résultats [17]. Dans une évaluation entièrement manuelle, toutes ces étapes sont réalisées par des experts humains. Les évaluations semi-automatiques mettent l'accent sur l'utilisation d'ordinateurs pour automatiser certaines tâches du processus [36]. Cependant, la participation d'humains pour effectuer certaines activités est toujours nécessaire. En général, les travaux portant sur l'évaluation semi-automatique introduisent des outils d'aide à l'évaluation par l'automatisation de la phase de calcul des résultats et la fourniture d'interfaces utilisateur pour télécharger des preuves d'évaluation ou des résultats partiels qui doivent être agrégés pour obtenir un résultat final. Les travaux proposés par [27-30] sont des exemples de tels outils logiciels. La génération automatique de rapports d'évaluation est une caractéristique commune de ces outils. Enfin, les évaluations contextuelles s'appuient sur des systèmes contextuels [37] qui se concentrent sur l'utilisation de capteurs physiques ou virtuels [37] pour recueillir des données sur l'entité évaluée. Les données collectées sont ensuite transmises à un gestionnaire de contexte capable de traiter et de raisonner en utilisant les données pour obtenir une représentation modélisée du contexte. Ce contexte, qui est défini dans [38] comme "toute information qui peut être utilisée pour caractériser la situation d'une entité", est ensuite distribué aux consommateurs intéressés [37]. Il existe différentes façons de traiter les évaluations contextuelles. Pour les évaluations qualitatives, par exemple, les techniques de Process Mining [39, 40] peuvent être employées pour utiliser des écouteurs qui pourraient être connectés aux systèmes de planification des ressources de l'entreprise (ERP) pour collecter les journaux du système afin de dériver des modèles de processus d'entreprise qui peuvent être ensuite analysés pour évaluer le processus. Un traitement peut également être effectué (sur les journaux d'événements ou sur les modèles obtenus, en utilisant la connaissance du domaine par exemple) pour obtenir des modèles de processus d'affaires structurés dans un but plus spécifique ou avec un certain caractère [31-33]. Pour les évaluations quantitatives, les évaluations contextuelles pourraient être basées sur l'utilisation de capteurs percevant un environnement industriel afin d'obtenir des données en temps réel qui peuvent ensuite être prétraitées et présentées aux parties prenantes au moyen d'outils de visualisation [34, 35] afin de prendre des décisions.

Les approches mentionnées précédemment font partie des différentes initiatives de recherche visant à améliorer le processus d'évaluation en proposant une certaine stratégie, méthodologie, artefact ou technologie. Ces approches sont généralement axées sur des scénarios et des actions concrètes. En outre, si l'on considère le niveau d'automatisation du processus d'évaluation, il existe un fossé entre l'évaluation semi-automatique et l'évaluation contextuelle. La première semble fortement axée sur l'automatisation de la phase de calcul des résultats de l'évaluation, tandis que la seconde est centrée sur l'automatisation des activités d'acquisition et de modélisation des données. Cette lacune représente l'absence d'approches d'évaluation entièrement automatisées capables d'entrer des données et de fournir des résultats de manière autonome afin de réduire l'effort humain et de fournir des résultats d'évaluation fiables. Dans ce sens, il y a un besoin pour un cadre holistique capable de considérer l'automatisation des évaluations d'une perspective de bout en bout. Ce cadre pourrait être utilisé par les chercheurs et les praticiens de l'industrie comme guide pour le développement de systèmes d'évaluation capables d'effectuer des évaluations entièrement automatisées, comblant ainsi le fossé entre l'automatisation de la collecte des données et les phases de détermination des résultats des évaluations. En outre, le cadre doit être suffisamment générique pour pouvoir guider la conception et le développement de systèmes d'évaluation indépendamment des algorithmes ou des ressources technologiques qu'ils utilisent. En ce sens, la généricité présente un ensemble d'avantages, par exemple, elle permet au cadre d'être instancié pour de multiples domaines en fournissant des

descriptions claires et simples de ses éléments et de la fonctionnalité de base du système à développer, sans avoir besoin d'une connaissance spécifique du domaine [41], il favorise la réutilisation des composants du système développé dans différentes applications [42], il est indépendant des ressources technologiques, ce qui donne aux concepteurs de systèmes la liberté de choisir la technologie qu'ils considèrent comme la mieux adaptée à leur problème, et il permet d'analyser si un système d'évaluation précédemment développé pour un certain domaine d'application a des éléments qui agissent comme des instances de certains éléments du cadre en utilisant une certaine mesure de similarité [43], ce qui évite de réinventer des technologies, des approches ou des méthodes.

Il convient de noter qu'un système d'évaluation développé à l'aide du cadre ne doit pas seulement être capable d'automatiser certaines tâches de calcul ou de consommer et de modéliser des données contextuelles, mais il doit englober un ensemble de capacités permettant d'assurer des évaluations efficaces de bout en bout. En ce sens, les capacités des systèmes intelligents telles que la connaissance, le raisonnement, l'apprentissage, la communication ou la perception [44] pourraient servir à cette fin, puisqu'elles leur permettent de percevoir en permanence un certain environnement, de faire des déductions concernant l'état de cet environnement, de communiquer entre les éléments du système et d'exécuter des actions spécifiques si nécessaire [44]. Par conséquent, le cadre pourrait inclure ces capacités pour guider le développement de systèmes d'évaluation intelligents qui pourraient être capables d'effectuer des évaluations automatisées de bout en bout. De plus, le concept d'intelligence dans le contexte de l'évaluation d'entreprise pourrait non seulement impliquer que le résultat de l'évaluation puisse être défini correctement par le système sans implication humaine, mais aussi que certaines actions puissent être prises une fois le résultat calculé afin d'initier instantanément le processus de transformation vers la réalisation de l'état To-Be. En effet, des mécanismes permettant d'exécuter automatiquement des actions pour modifier l'environnement et réagir à de nouveaux états de cet environnement pourraient également être utilisés une fois le résultat de l'évaluation obtenu [45], fermant ainsi un cycle composé de la détection de l'environnement, de la réflexion et de la décision sur les actions possibles à entreprendre en fonction de son état, et de l'exécution effective de ces actions. Bien que cela ne soit pas nécessaire au cours d'une évaluation puisque les évaluations se concentrent uniquement sur l'établissement d'un diagnostic de l'entité évaluée, il s'agit d'une capacité inhérente aux systèmes intelligents et elle pourrait également être prise en compte par les concepteurs de systèmes s'ils le jugent nécessaire. D'autre part, non seulement les éléments externes de l'environnement pourraient être modifiés, mais aussi les composants internes du système d'évaluation. Plus précisément, la capacité d'apprentissage d'un système intelligent pourrait permettre de s'adapter à de nouveaux défis ou d'améliorer ses propres performances, de manière à fournir des résultats d'évaluation plus fiables au fur et à mesure que le système évolue en utilisant ou en visualisant de nouvelles données.

Compte tenu du contexte et de la motivation de la recherche, cette thèse vise à combler l'absence d'un cadre générique capable de guider la conception et le développement de systèmes automatisés d'évaluation de bout en bout dans les entreprises. L'objectif de cette thèse est de proposer un cadre pour effectuer des évaluations intelligentes d'entités d'entreprise : le Smart Assessment Framework (SAF). L'hypothèse de ce travail est qu'un cadre conceptuel basé sur les concepts de systèmes intelligents peut fournir un support efficace pour le développement de systèmes permettant d'effectuer des évaluations automatisées dans les entreprises.

La définition d'un tel cadre implique une série de défis qui doivent être relevés : (1) Une analyse du concept d'intelligence ainsi que des caractéristiques des systèmes intelligents; (2) Le cadre d'évaluation de l'intelligence doit être conçu en tenant compte à la fois des caractéristiques des systèmes intelligents et des activités du processus d'évaluation; (3) La validation du cadre doit être

effectuée en tenant compte de différents scénarios d'évaluation. Les scénarios pourraient être basés sur des cas d'utilisation du monde réel et définis de différentes manières pour couvrir divers aspects tels que les preuves d'évaluation qui doivent être traitées (structurées, non structurées, etc.), la caractéristique à analyser (maturité, capacité, interopérabilité, risque, etc.) et le type d'évaluation (qualitative ou quantitative). Le cas d'utilisation abordé dans cette thèse se concentre sur les scénarios d'évaluation des capacités des processus métier. Les processus métier sont essentiels car ils comprennent l'ensemble des activités ou des fonctions métier qui sont exécutées selon une certaine séquence afin d'obtenir des résultats qui apportent de la valeur à l'organisation [46]. L'intérêt de comprendre l'état des processus métier et de les améliorer constamment réside dans le fait que la plupart des problèmes liés à l'efficacité et à l'efficacité d'une organisation trouvent leur origine dans des processus non maîtrisés [22].

Compte tenu de l'objectif de ce travail et des défis de recherche énumérés précédemment, l'accent est mis sur la réponse à la principale question de recherche (QR) : *Quels sont les éléments qui composent un cadre d'évaluation d'entreprise intelligent?*

Pour répondre à cette question, l'objectif principal de cette thèse est de répondre au besoin d'un cadre fondé sur les capacités des systèmes intelligents afin de guider le développement d'outils, d'approches et de méthodologies pour effectuer des évaluations d'entreprises plus intelligentes. Dans ce sens, le cadre doit être d'abord correctement défini et ensuite validé. Les objectifs secondaires de ce travail, qui sont présentés ci-dessous, sont conçus dans le but de répondre à ces deux exigences : (1) Concevoir un cadre pour effectuer des évaluations intelligentes des entités de l'entreprise; (2) Valider le cadre en considérant différents scénarios d'évaluation.

Le premier objectif spécifique est lié à la définition du cadre. Par souci de clarté, une approche de modélisation conceptuelle est utilisée pour la définition du cadre, ce qui permet d'améliorer la compréhension de son architecture en rendant visibles ses caractéristiques pertinentes et les relations complexes entre ses éléments [47]. Ce thèse propose le Smart Assessment Framework (SAF), un cadre conceptuel qui a été conçu pour servir de base à la développement et mise en œuvre de systèmes d'évaluation dotés de capacités intelligentes telles que le raisonnement, l'apprentissage et la perception des données. Les éléments du SAF sont fondés sur des concepts qui apparaissent dans la littérature scientifique, les normes internationales, et ils sont spécialement basés sur les caractéristiques des systèmes intelligents.

En ce qui concerne la validation du SAF, cette thèse se concentre sur l'évaluation qualitative des processus d'affaires. Plus précisément, des scénarios portant sur l'évaluation des capacités d'un processus métier de gestion des échantillons chimiques sont ciblés. La validation est donc basée sur l'implémentation de systèmes chargés de supporter les évaluations de ces scénarios. Ceci est réalisé en instanciant les éléments définis par le cadre. La principale différence entre les scénarios est le type de preuves utilisées pour l'évaluation. En ce sens, deux types de preuves sont considérés pour les implémentations : les descriptions textuelles des processus et les modèles de processus métier. En ce qui concerne ce dernier type de preuve, un aspect pertinent lors de l'évaluation des capacités utilisant des modèles d'entreprise est que les informations qui peuvent être trouvées dans ce type d'artefacts dépendent du langage de modélisation qui est utilisé pour les concevoir. En ce sens, comprendre quelles exigences d'évaluation peuvent être satisfaites par les modèles conçus avec un certain langage est une étape essentielle lors du développement d'un système conçu pour le traitement automatique de ces modèles. C'est une question qui est également abordée dans cette thèse.

La première implémentation du SAF est liée à l'utilisation des descriptions de processus au format texte pour faire les évaluations. Cette implémentation est basée sur une approche hybride qui combine l'utilisation de techniques de Deep Learning pour le traitement du langage naturel et des Ontologies

pour la représentation et l'inférence des connaissances. Concrètement, une architecture pour un réseau de neurones de type Long Short-Term Memory a été utilisée pour traiter le texte, et une Ontologie nommée Process Capability Assessment Ontology a été proposée pour calculer les résultats de l'évaluation et disposer d'une base de connaissances décrivant les caractéristiques de chaque nouvelle évaluation.

La deuxième implémentation est liée à l'utilisation des modèles de processus pour faire les évaluations. L'implémentation utilise un réseau de neurones de type Graph Convolutional Neural Network (GCN). Le GCN a été entraîné pour agir en tant que classificateur sur la qualité des processus en utilisant les modèles comme données d'entrée.

Comme première étape pour le traitement des modèles des entreprises et avant l'application du GCN, la thèse propose aussi une approche systématique basée sur le paradigme Goal Question Metric et la méthode Pseudo-Requirement Graph pour définir quelles exigences peuvent être satisfaites ou évaluées lors de la réalisation d'évaluations en utilisant des modèles d'entreprise.

Cette thèse est destinée à servir de source de contributions sous deux angles : conceptuel et pratique. Le SAF est composé d'une structure conceptuelle artisanale qui permet à ses éléments d'être instanciés pour définir des systèmes d'évaluation à utiliser dans différents contextes en tirant parti des capacités intelligentes. De même, il fournit des descriptions détaillées de ces éléments et de leurs relations, afin de fournir des directives claires pour leur instanciation ultérieure. D'un point de vue pratique, les implémentations du SAF (pour le texte et le traitement des modèles d'entreprise) introduites dans cette thèse sont entièrement fonctionnelles et elles peuvent être directement utilisées dans des scénarios industriels pour effectuer des évaluations. En outre, elles peuvent être utilisées dans un contexte académique, étant donné qu'elles sont également conçues pour être exploitées comme des bases pour le développement d'autres approches qui peuvent être construites comme des expansions ou des compléments des implémentations originales.