



**HAL**  
open science

## A framework for assessing capability in organisations using enterprise models

Marcelo Romero, Wided Guédria, Hervé Panetto, Béatrix Barafort

### ► To cite this version:

Marcelo Romero, Wided Guédria, Hervé Panetto, Béatrix Barafort. A framework for assessing capability in organisations using enterprise models. *Journal of Industrial Information Integration*, 2022, 27 (100297), 10.1016/j.jii.2021.100297 . hal-03379889

**HAL Id: hal-03379889**

**<https://hal.science/hal-03379889v1>**

Submitted on 15 Oct 2021

**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.

# A Framework for Assessing Capability in Organisations Using Enterprise Models

Marcelo Romero<sup>a,b</sup>, Wided Guédria<sup>a,b</sup>, Hervé Panetto<sup>b</sup> and Béatrix Barafort<sup>a</sup>

<sup>a</sup>Luxembourg Institute of Science and Technology (LIST), 5, Avenue des Hauts-Fourneaux, L-4362, Esch-sur-Alzette, Luxembourg

<sup>b</sup>Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France

---

## ARTICLE INFO

### Keywords:

Enterprise Modelling  
Capability Assessment  
Requirements Engineering  
Business Process Model and Notation  
ArchiMate

## ABSTRACT

Organisational assessments provide a view of the state of different aspects of an enterprise so as to understand its strengths, weaknesses and possible improvement paths. A relevant element used to perform assessments is the assessment framework, which contains the requirements that the entity must fulfil as well as other aspects. On the other hand, another essential element during an assessment is the Evidence, which is any type of data reflecting the state of the assessed entity. Among several possible assessment evidences, such as documents, e-mails, interview extracts, and questionnaires, Enterprise Models represented through a specific Modelling Language can serve as such, since they allow to describe diverse organisational aspects. Thus, they can provide useful information to be considered when performing an assessment. Notwithstanding, the analysis of the models is demanding in terms of time and resources, and reducing the effort to perform such analysis could improve the assessment process. Thus, a possible manner of improving the analysis process is to understand beforehand which type of information can be found within an enterprise model taking into consideration the modelling language that has been used to represent it and the list of requirements defined by the assessment framework used to perform the assessment. In this sense, this work introduces a method 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. We ground our approach on a Requirement Decomposition task following the Pseudo-Requirement Graph method, and the matching between requirements and modelling language elements following the Goal Question Metric paradigm. To validate our method, we perform separate evaluations of the BPMN and ArchiMate languages, to identify the set of requirements from the ISO/IEC 33020 international standard that can be answered through models that are defined using each language. We also introduce a case study based on the analysis of a concrete business process, following the proposed approach.

---


## 1. Introduction

The challenges faced by enterprises in a daily basis, such as mergers, acquisitions, novel technologies, and highly dynamic market, foster them to be in a constant cycle of renewal. Enterprises are systems that are in motion [1] to address those challenges, which could be in the form of adaptation to market needs, alignment of the operations to novel strategies, implementation of disruptive technologies, among others. In this context, motion or changing initiatives must be directed towards the accomplishment of objectives, goals and regulations. However, to perform these changes, enterprises require an objective view of their state so as to understand their strengths and weaknesses, and identify possible improvement paths.

Assessments are performed to address this issue, they allow to obtain a view of different enterprise aspects such as enterprise interoperability [2], project agility [3], and process performance [4]. Assessments can be qualitative or quantitative. Considering their purpose, they could be descriptive, based on providing a view of a specific entity, prescriptive, based on providing improvement recommendations, and comparative, focused on performing benchmarking between industries or regions [5]. In general, to perform an assessment, an assessment framework [6] is used, which provides a schema to be followed by the assessors to effectively perform an assessment.

A specific type of assessment is Process Capability Assessment [7], which is focused on measuring the extent to which a Process achieves certain specifications. There are five steps that are followed during Capability Assessment: planning, data collection, data validation, results determination, and results presentation [8]. Among these steps, a

---

 nmarceloromero@gmail.com (M. Romero); wided.gue@gmail.com (W. Guédria)  
ORCID(s): 0000-0002-8673-3744 (M. Romero)

particularly critical task is data collection, since the collected data must reflect the reality of the assessed entity in order to provide relevant and reliable assessment results. In this step, 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 [9] 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 [10]. In this context, an enterprise could have one or more models that can be considered as evidence during an assessment. 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 the analysis could be valuable to improve the overall assessment process. In this sense, a possible manner of improving the analysis process is to understand beforehand which type of information can be found within an enterprise model taking into consideration the modelling language that has been used to represent it and the list of requirements defined by the assessment framework used to perform the assessment.

To tackle this issue, this work aims at defining a methodology to identify the requirements, devised by an assessment framework, that can be answered through the analysis of the elements defined using a specific modelling language. In short, we aim at providing the means to evaluate the ability of a modelling language to provide the necessary modelling elements to identify and verify a set of assessment requirements. We consider that the proposed method will aid assessors at identifying, through a simple process, the requirements that can be answered using enterprise models, available for organisational assessment, by taking into consideration the modelling language in which the model was defined. This possibility will allow assessors to reduce time when reviewing evidences during assessments, allowing them to consider only those requirements that are visible through the enterprise model that is being analysed, taking into consideration the modelling language used to define the model. As proof of concept, we perform a method validation based on the analysis of the capability of Business Process Modelling Notation (BPMN) [11] and ArchiMate [12] languages to support requirements defined by the ISO/IEC 33020 international standard for process assessment [6]. Moreover, we also introduce a case study with the objective to show that the proposed approach can also be used to analyse concrete enterprise models, given a list of requirements defined by some assessment framework.

The contributions of this work are: (1) to propose 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) to present 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) to describe the possibilities of BPMN and ArchiMate to provide assessment results following the ISO/IEC 33020 measurement framework for assessment of process capability, (4) to perform an assessment of a real-world business process through the analysis of enterprise models describing its characteristics, for such purpose we follow the method presented in this work. Note that the proposed methodology has two main uses or levels of application: the first level refers to the possibility to measure the ability of a modelling language to evaluate assessment requirements; the second level is related to the proper verification of those requirements in a concrete model, which is represented using a language able to answer them. Contributions (2) and (4) show examples of the application of the proposed approach for the first and second levels, respectively.

This paper is organised as follows. Section 2 presents concepts that are addressed throughout the paper, along with the related work. Section 3 details our proposed approach to perform the modelling language analysis for supporting requirements from an assessment framework. The method validation devised to test the methodology is presented in Section 4. Section 5 presents a case study that consists in analysing two enterprise models following our approach. Finally, Section 6 enumerates the conclusions from our work and introduces future research perspectives.

## 2. Background and Related Work

This section presents basic concepts that are addressed throughout this paper. First, we introduce the notion of assessment in enterprises with specific focus on process capability assessment. A description of concepts regarding enterprise modelling is then presented, and we briefly introduce some common modelling languages that are used to represent different enterprise entities. Finally, we provide a description of works that aimed to develop or apply methods to analyse the capability of modelling languages to support maturity or capability assessment in enterprises.

## 2.1. Maturity and Capability Assessment

An assessment is the act of estimating or deciding the amount, value, quality, or importance of a certain entity [13]. In general, an assessment is composed of the following activities: planning and preparation, data collection and validation, results determination, and results presentation.

Assessments can be classified according to different aspects. Considering the measuring mechanism, there are two types: qualitative and quantitative. Qualitative assessments are mostly based on the use of subjective measures to evaluate the process, generally they are based on a rating scale composed of linguistic variables (e.g. “Regular”, “Good”, “Optimised”) [14]. On the other hand, quantitative assessment methods define numeric values to provide a final representation of the quality of the assessed entity. Quantitative assessment is more commonly focused on measuring performance. Regarding the openness of the assessed entity, the literature [15] considers two types of assessment, which were inherited directly from the general system theory [16]: white-box and black-box methods. Assessments can also be classified considering the time of application. The *a priori* assessment, also known as *ex-ante* evaluation, is applied to assess the potential of the assessed entity. 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. 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. 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.

Finally, a common classification perspective to group assessments types considers if they are focused on maturity levelling. In this sense, maturity assessment is based on the use of models that reflect aspects of an assessed entity in terms of their capability, through the definition of qualitative attributes that are used to categorise it into one or more clearly defined classes [17]. Specifically for business processes, maturity assessment represents an effective way to measure them within an organisation [18]. The capability, on the other hand, represents the extent to which a single process is explicitly or consistently deployed by an organisation; if a list of processes is considered, the assessment is then measuring the maturity of the organisation [7]. In this sense, maturity models conceptually represent levels of increasing capability of a maturing element, which can be qualitative or quantitative, in order to evaluate its advances considering some defined areas [17]. Since they use levels as evaluation result, maturity assessment approaches are considered levelling methods [15]. 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 (or capability level for single processes), instead they are specialised to a particular type of system [15]. Non-levelling methods aim at addressing specific gaps inherent to maturity and capability methods. One relevant issue that must be considered is that maturity models do not imply a performance measurement: a high maturity level does not always imply a high performance [7]. Thus, non-levelling methods are more suited for this task.

Figure 1 presents relevant elements of an assessment, specialised to reflect Capability assessment of Business Processes. The elements are based on the meta-model presented in [19], and it is not intended to reflect a full view of the assessment process, the objective is rather to show how capability assessment is linked to the requirements defined by some assessment framework. In this work, we specifically consider the Requirements defined by the ISO/IEC 33020 [6] assessment framework for business process capability assessment. Note that, in the specialisation we are considering Process Capability as the Quality Characteristic to be assessed, since the assessed entity is a single Business Process. Regardless of the assessment scope, in all cases there is a list of Requirements that define the Quality Attribute to be evaluated during the assessment.

As shown in Figure 1, a relevant component of the assessment is the assessment framework, which is a document that provides a schema to be used for characterising some qualitative or quantitative characteristic of an assessed entity, it includes a list of generic requirements that the entity must fulfil. It also includes the results calculation and determination methods, which are not shown in the model for the sake of simplicity. In this context, the requirements define a list of criteria that the Assessed Entity must fulfil in order to reach a certain level of quality or compliance to some regulation or standard. For business processes, an assessment framework often clarifies relevant elements of the process including actors, goals, objectives, requirements and key performance indicators in addition to the elements described previously.

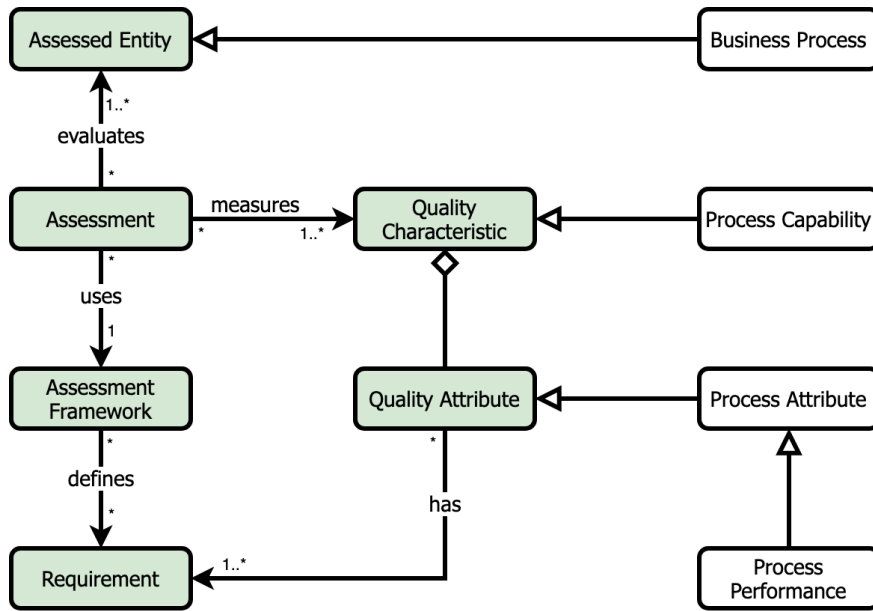


Figure 1: An overview of the elements of an assessment. The model is specialised for Business Process Capability Assessment.

A well-known maturity model that includes an assessment framework is the Capability Maturity Model Integration (CMMI) [20]. It is a *de facto* standard that defines global best practices and a list of maturity levels to define the maturity of an organisation. The official assessment framework to guide CMMI assessments is the Standard CMMI Appraisal Method for Process Improvement (SCAMPI), which provides guidelines to perform appraisals grouped into three categories according to the level of formality of the evaluation: SCAMPI A, SCAMPI B, and SCAMPI C. On the other hand, the ISO/IEC 33020 [6] standard is a Process Assessment Measurement Framework that defines a list of process capability levels in which each level groups one or more Process Attributes that define a list of Indicators (defined in ISO/IEC 330xx standards as Process Attribute Outcomes) to evaluate a business process, which are requirements themselves. It also provides a result calculation method. Note that this naming and conceptualisation of an assessment framework could change depending on the maturity model or standard used as point of reference. For the purpose of this research work, we use the concepts proposed by the ISO/IEC standard 330xx series since it is a well-known and established international standard that is widely used in the literature.

## 2.2. Enterprise Modelling

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 [21]. 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 [21]. 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 [21].

The elements of a modelling language are graphically represented in Figure 2. 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 [21]. The method essentially provides help to modellers who use one or more modelling languages and describes the activities to be performed for the development and validation of a model [22]. 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.

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 [9] as a means of understanding and analysing the enterprise [23]. The objective of enterprise models is to provide organisations the capability to perform enterprise design, analysis and operation with models as foundation, following

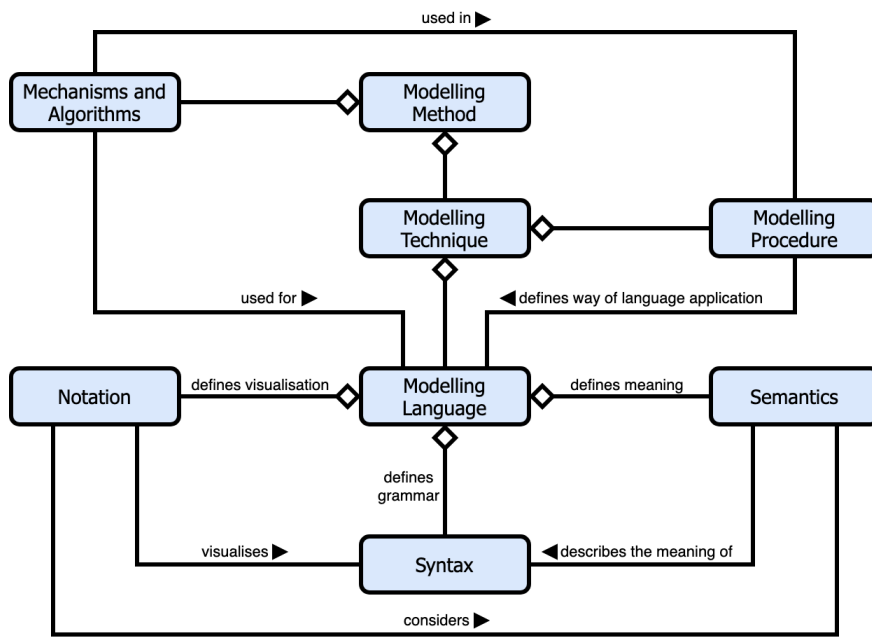


Figure 2: Elements of a modelling language and other related aspects (adapted from [21]).

a model-driven approach [9]. Moreover, enterprise modelling is considered as an efficient tool to represent and externalise the knowledge of companies with the purpose of understanding and analysing their operations, and designing novel systems from different perspectives: functions, processes, decisions, resources, information technology [24]. Thus, the purpose of an enterprise model is to reflect relevant features and characteristics of a system that is the objective of [10]. In this sense, a model is an abstraction of one or more aspects of a real-world system, situation or phenomenon [10]. Hence, it serves as a approximation of the reality at a given point of time.

In this sense, 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 [25].

Enterprise Models are expressed using semi-formal Enterprise Modelling Languages (EML) [10] 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 [10]. Enterprise models can be processed by computers and they can have static or dynamic nature, depending on the purpose of the model [10]. 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 [10, 26].

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 [10], which were summarised from diverse sources [27, 28, 29, 30]:

- 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 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 of the model devised for a specific population of users given by the details of the system to be taken into consideration.
- 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) [31], 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 [32]. These languages cover several aspects such functional modelling (IDEF0), information modelling (IDEF1), dynamic modelling (IDEF2), data modelling (IDEFIX), among others. Examples of languages that are more specific for the business context are Business Process Model and Notation (BPMN) [11], which provides a graphical representation of business processes in a business process model; and ArchiMate [12], 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) [33] 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 [34], 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 [35] with the objective to serve as an interface between enterprise modelling languages, tools, and frameworks [36]. On the other hand, considering a Computer-integrated Manufacturing (CIM) context, the GRAI method [37] 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 [38]. Another modelling approach focused on enterprise integration is CIMOSA [39], which structures a CIM system as a set of processes that intercommunicate and are executed by a set of functional entities [40]. 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 [41]. 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 we foster the reader to refer to the works by [24] and [10].

## 2.3. Modelling Languages

This work is specifically focused on two modelling languages: BPMN and ArchiMate. In this section we present a brief description of each of them.

### 2.3.1. Business Process Model and Notation (BPMN)

BPMN 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 [42]. BPMN defines a Business Process Diagram (BPD) based on a flowcharting technique that allows to create graphical models describing business processes [43].

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 [42]. The Core layer contains three sub-packages: (1) **Foundation**, including the fundamental constructs needed for BPMN, (2) **Service**, including the fundamental constructs needed for modeling 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 [42].

### 2.3.2. 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 [12]. Figure 3 presents the organisation of the layers within the scope of the ArchiMate framework [44].

The layered-structure of the framework is intended to represent enterprise elements with different levels of abstraction. 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.



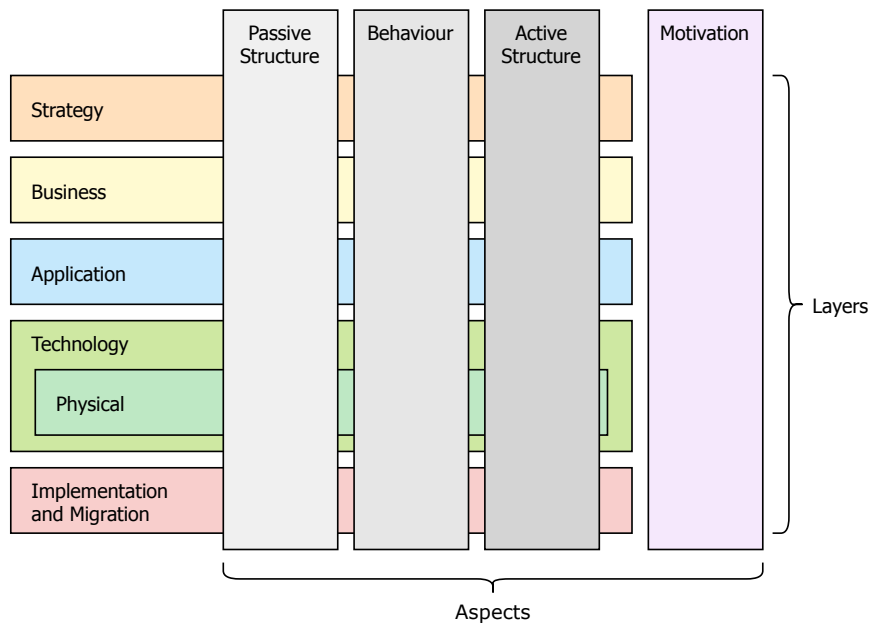


Figure 3: Framework of ArchiMate (adapted from [12]).

- **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 represented as columns in Figure 3:

- **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 this language. For an extended description of ArchiMate, the reader may refer to the ArchiMate specification 3.1<sup>1</sup>.

## 2.4. Related Work

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 an assessment methodology. The work by [45], 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 after to obtain

<sup>1</sup><https://pubs.opengroup.org/architecture/archimate3-doc/toc.html>

the assessment results. In [46], 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)<sup>2</sup>. 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, [47] 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 [48], which contains definitions of BPMN concepts and it was extended to contain dependency relations. In [49], the authors proposed an enterprise architecture meta-model of ISO 27001 [50] and its mapping to COBIT 5<sup>3</sup> 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 [51] 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 above 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 assessment. Likewise, some works have also aimed at evaluating modelling languages or techniques that are able to automatically provide models using some kind of input data. In [52], for instance, the potential of Process Mining [53] to support software process assessment and improvement approaches has been addressed. Generic Practices of the Capability Maturity Model Integration [20] 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, [54] 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.

To the best of our knowledge, there is no 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. 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. This work deals with these steps, proposing a structured approach based on a requirements decomposition method [55], and a requirement and modelling elements matching approach based on the Goal Question Metric (GQM) paradigm [56]. 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 tool that can be used for the analysis of different modelling languages and assessment frameworks. In this work, we use the ISO/IEC 33020 assessment framework and the BPMN and ArchiMate languages 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 defined in the standard 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 languages.

### 3. Methodology

This work 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

<sup>2</sup><https://gdpr.eu>

<sup>3</sup><https://www.isaca.org/resources/cobit>

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 to the modelling language used to represent them, thus improving the verification step of the available models during an assessment. Figure 4 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 [31] to visually represent the elements of the meta-model.

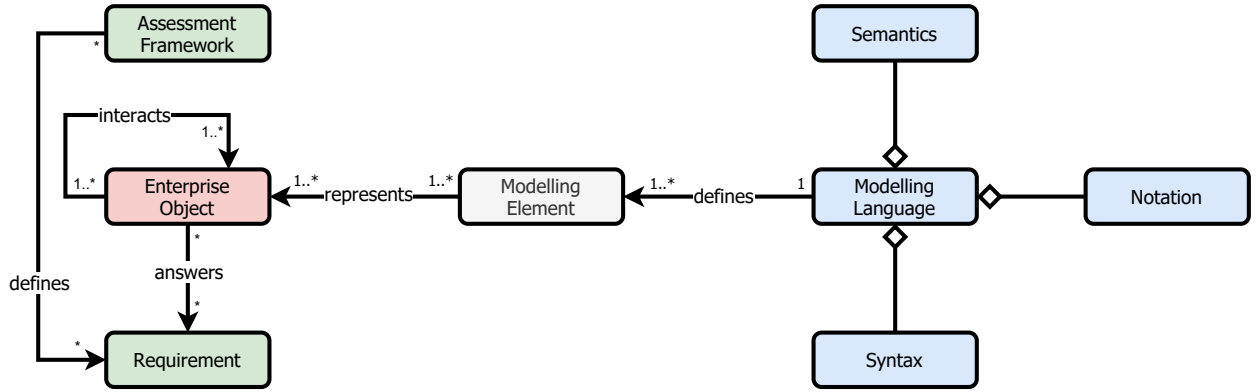


Figure 4: Relationship between the requirement of the assessment framework and the Element of the modelling language.

Note that the meta-model is composed of the main elements presented in figures 1 and 2. The main element of the meta-model is the Enterprise Object (EO) and the interactions between EOs. In the context of our 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 [57] 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 our 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 this 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 [58].

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 act 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 presents this process graphically.

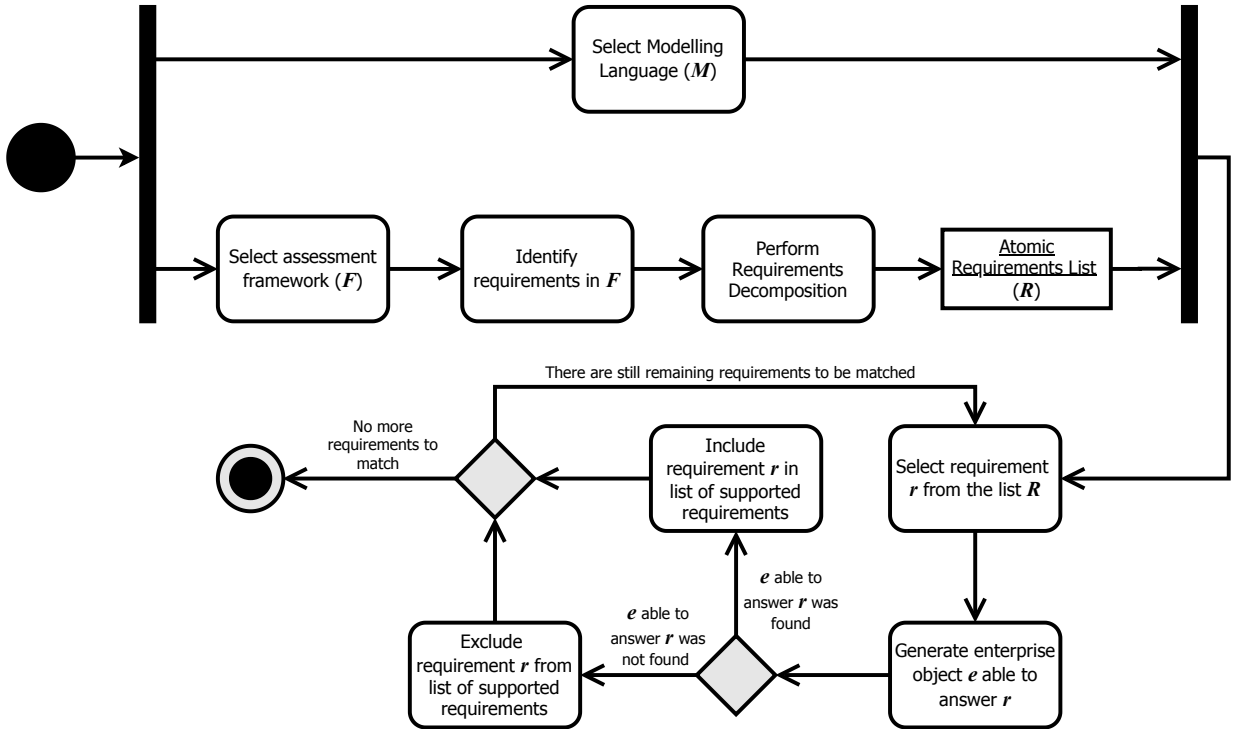


Figure 5: Overview of the evaluation process.

Among the steps of the proposed methodology, we identify two particularly relevant tasks to provide satisfactory matching between requirements and modelling elements: (1) Requirements Decomposition, and (2) Requirement and modelling language matching. In the following sections we describe those tasks in detail.

### 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 our approach, they can be matched with EOs that are able to answer them.

To perform Requirements Decomposition we rely on the approach proposed in [55], which was successfully applied to different contexts in the literature such as software analysis [59] and interoperability assessment [58]. 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 6 presents an overview of the main tasks of the process.

The process initiates with the selection of the pseudo-requirement to be analysed, which is then decomposed into sub-requirements (there are two sub-requirements in the figure to serve as example). 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\* [60], an extension of CTL (Computation Tree Logic) [61]; 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

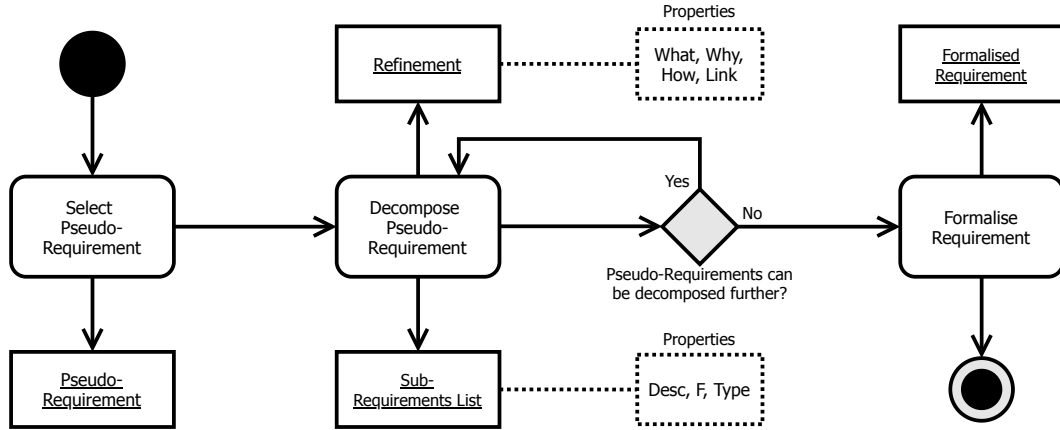


Figure 6: Overview of the requirements decomposition and formalisation through the Pseudo-Requirement Graph-based approach. We use dotted lines and boxes to describe the properties of the output artefacts of the activities of the model.

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  $\phi$  has to hold at the next state if **X** is applied to it ( $\mathbf{X} \phi$ ). The later means that for two propositions,  $\phi$  and  $\psi$ , that are connected through **U** ( $\phi \mathbf{U} \psi$ ),  $\phi$  has to hold at least until at some position  $\psi$  holds. This adds the possibility to formalise temporal conditions, which are often present in requirements definitions.

Figure 7 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 we derive a new requirement 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 **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 **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, we apply a refinement 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  $Rq26$ .

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. We foster the reader to refer to the original paper [55] for more details regarding the Pseudo-Requirement Graph approach. For simplicity, we refer to the atomic requirements obtained through the approach described in this section simply as requirements in the following sections.

### 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 [56], which is a systematic approach for defining and evaluating a set of goals through some measurement mechanism. The paradigm was used in [52] to assess the Generic Practices of the Capability Maturity Model Integration [20] that can be addressed through the use of Process Mining methods [53]. This approach is

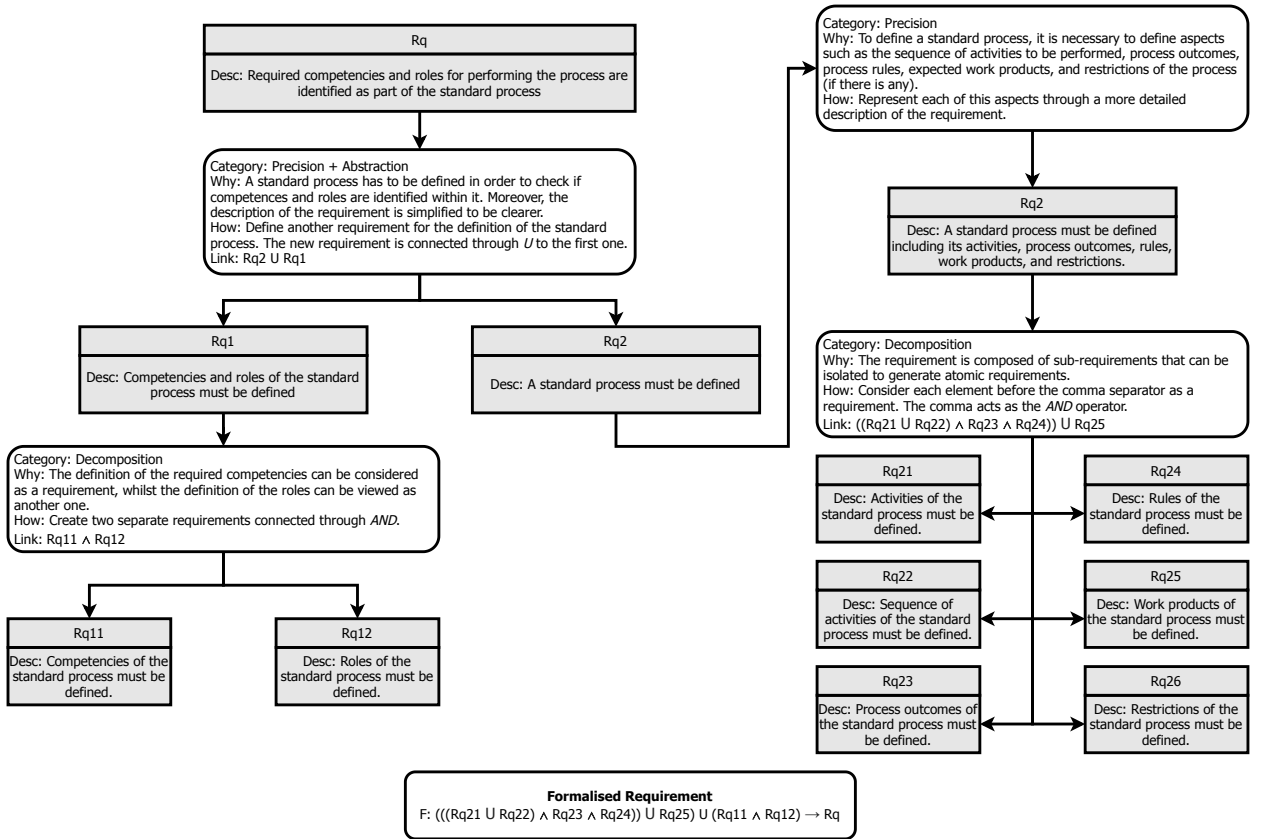


Figure 7: Example of the decomposition of a requirement  $R_q$ .

considered as a starting point of the requirement and modelling elements matching step of our 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 [56]. 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 [52], 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 8 graphically presents the relationships described before.

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 our method since it depends on the specific  $F$  used and on the manner that the results will be presented to the stakeholders.

#### 4. Method Validation: Analysing BPMN and ArchiMate considering ISO/IEC 33020

To validate our approach, we followed a strategy based on analysing the adequacy of the BPMN and ArchiMate modelling languages to answer requirements from the ISO/IEC 33020 standard Measurement Framework [62] for

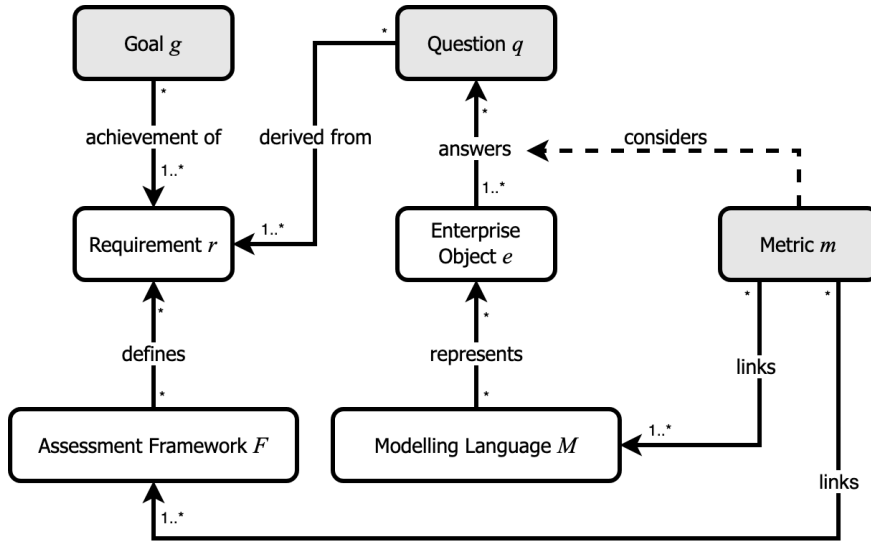


Figure 8: The Goal Question Metric paradigm within the scope of our approach.

process capability assessment, which is described in Section 4.2. Note that the version of the standard that was chosen is the one from 2015, which was selected considering the experience of the authors at performing assessments in the industry using this particular version. Thus, it is a well-known document for the assessment team. However, the approach introduced in the previous section is intended to be independent from the version or type of assessment framework that is used, as long as it includes a set of requirements that must be checked during an appraisal. Some assumptions regarding the modelling languages used for the validation are introduced in Section 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. We present the list in Section 4.3 with the analysis of the results.

#### 4.1. Considerations

Following our proposed approach, we make five **Assumptions** regarding BPMN for our validation:

- We consider the elements of BPMN, version 2.0, without any extension.
- 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 are not explicitly related to the assessed process. This is, non-functional aspects of the process.

Note that the last assumption is perhaps the most complex and it requires a more detailed explanation. We make this assumption 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, we assume the nonexistence of such models and we assume only the existence of models explicitly reflecting the execution aspects of the assessed process.

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

- We consider elements of ArchiMate, specification 3.1, without any extension.
- The ArchiMate model 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. We consider elements from all aspects that can be used to represent a process.
- The ArchiMate model is a reflection of the reality. Hence, it presents the As-Is state of the process, not the To-Be.

#### 4.2. ISO/IEC 33020 Measurement Framework

ISO/IEC 33020 is an International Standard that defines a Measurement Framework for measuring the process quality characteristic of process capability [62]. 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 our methodology, the Indicators of the framework are analogous to the requirements that must be refined during the first step of the approach.

Table 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. We exclude Level 0 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.

<b>Id</b>	<b>Process Attribute</b>	<b>Indicators</b>
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
<b>Total</b>		<b>43</b>

Table 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.

Note that 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 paper, since our 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, we only use those from Levels 1 and 2 to validate the proposed approach, which we consider as sufficient to demonstrate the functioning of the method.

#### 4.3. Results and Discussion

We applied the methodology proposed in Section 3 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.



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 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:  $Rq21a1 \wedge Rq21a2 \Rightarrow Rq21a$ .

In the context of our 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 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 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, we analysed the capability of the modelling language *M* to support the requirements. For each atomic requirement *r*, we define a Goal *g*, which in our case always refers to the capability of the language to represent an EO related to *r*. After, we define the Question *q*, 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 3 and 4 show the results for some requirements considering the ArchiMate language. Table 3 presents a scenario in which the requirement is able to be answered by the language, whilst 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 shows the requirements extracted from each Indicator of the PA. It is worth mentioning that some atomic requirements

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

Table 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.

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

Table 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.

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  $Rq22b3$ ). This is 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 we aimed at answering 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 we obtained the result (defining if the requirement can be answered through some modelling element) for all requirements  $r$  in a list  $R$  originated from the Indicator, we aggregated them considering the aggregation function  $s$  presented in Equation 1.

$$s = \begin{cases} \text{Yes,} & \forall r : (r \text{ has } e) \wedge (e \text{ supported by } M) \\ \text{Partially,} & \exists r : (r \text{ has } e) \wedge (e \text{ supported by } M) \\ \text{No,} & \text{otherwise} \end{cases} \quad (1)$$

The Indicator is supported by the modelling language  $M$  if all requirements have EOs  $e$  that are supported by  $M$

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

( $s = Yes$ ) 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 9 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 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,

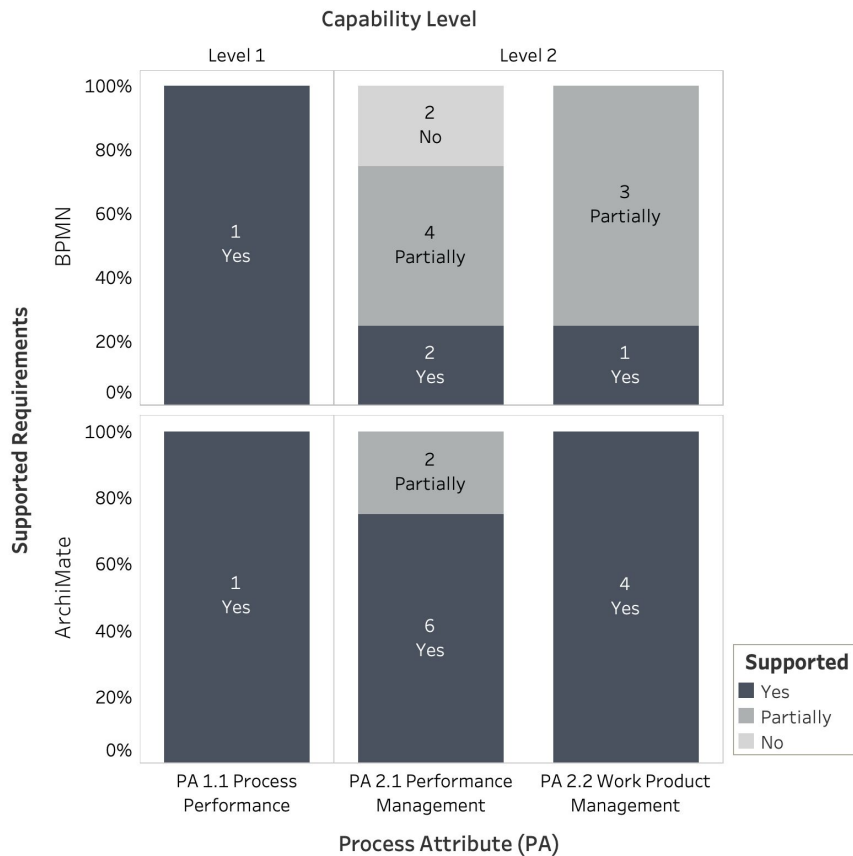


Figure 9: Results obtained through the analysis of the BPMN and ArchiMate modelling languages following the proposed methodology.

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 our 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. Nevertheless, 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. These results can be considered as expected ones, given that ArchiMate was originally devised to cover enterprise aspects from a holistic perspective. We argue that similar outcomes are likely to be obtained if modelling languages (and their integration) used in the context of multi-viewpoint modelling frameworks such as CIMOSA, GIM or ARIS are considered as objects of analysis. More experimentation could be performed in a future work to validate this hypothesis. On the other hand, 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. Case Study: Assessing Capability Level of a Business Process

The method validation presented in Section 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, we perform 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 by it. We specifically perform the assessment of one capability level for the process. During the analysis, for each Indicator we give a rating based on the scale defined in the ISO/IEC 33020, which is defined as follows: N - Not Achieved, P - Partially Achieved, L - Largely Achieved, F - Fully Achieved. It is worth mentioning that the assessment team is composed of two of the authors that analyse the model and define a rating for each Indicator unanimously. Note that the values of the ratings are subject to the experience and expertise of the members of the assessment team. In this sense, for all assessments it is expected that the ratings given by the assessors are as objective, impartial and transparent as possible.

The process to be assessed 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. In this section, we analyse the content of each model to assess if the process complies with the Indicators that are object of analysis. For this case study we make the following assumptions:

- 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 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 presented. Finally, we present a brief discussion addressing relevant aspects of the case study.

### 5.1. BPMN

The BPMN model is presented in Figure 10. 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.

Since it is already known which Indicators can be viewed through BPMN models, we only focus on them for the analysis. Moreover, for simplicity, we only consider Indicators from Capability Level 2, 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 4.

To analyse each Indicator, we use the available information obtained during the analysis of the modelling languages, presented in Section 4. Specifically, we use 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 assessors search for 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 6 presents an example of the information used by the assessors to provide the ratings for the Indicators derived from PA 2.1.

It is worth mentioning that when we analyse atomic requirements from PA 2.1, 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, we present the following definitions:

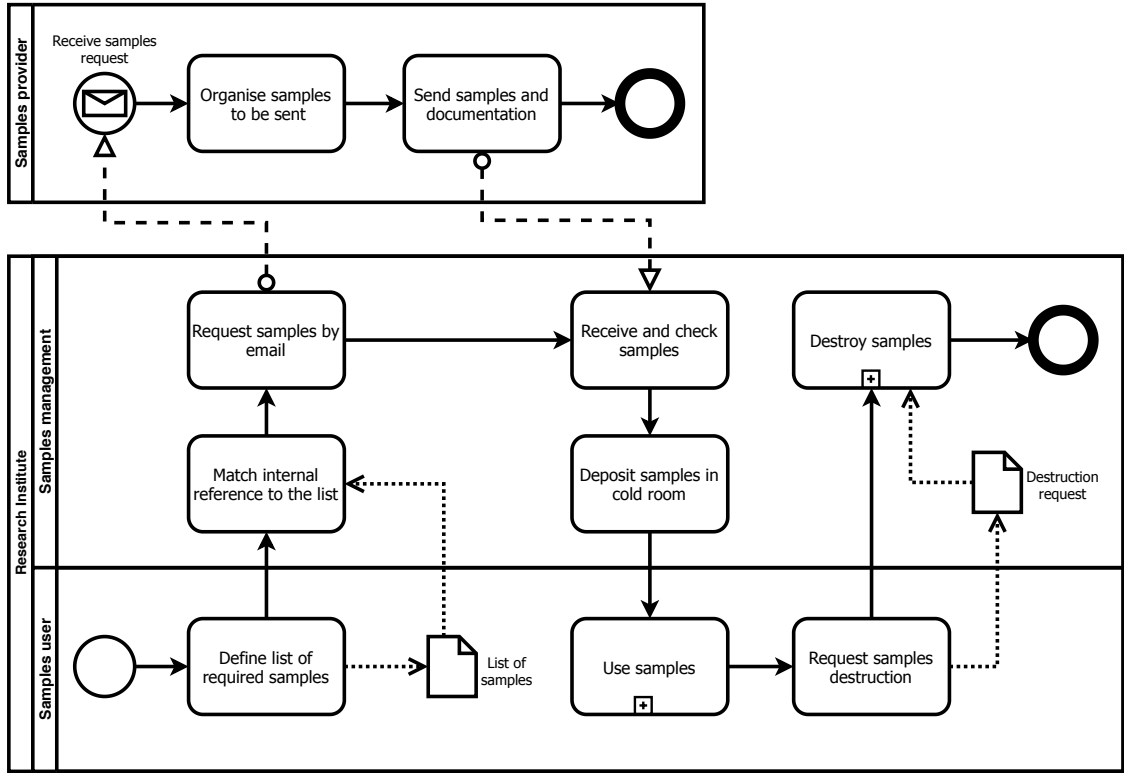


Figure 10: BPMN model describing the Samples Management Process.

- Quantitative Performance for SM Process:** we consider two main quantitative performance aspects, *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 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, the assessors conclude that there is a lack of sufficient monitoring activities for other aspects of the process, and that the description of the existing ones is rather superficial. The assessment team gives a rating of P - Partially Achieved 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 7.

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

Atomic Requirement (r)	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 6: Ratings provided 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.

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 7. Specifically, we identified three elements that are related to work products: 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. We consider that the definition and identification of the work products happens at the moment the work product is included in the enterprise model. Our 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

Atomic Requirement (r)	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 7: Ratings provided 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.

considered, the existence of separate requirements for the definition and identification of work products is therefore necessary. As stated before, the assessment team defined the same rating for *Rq22c1* and *Rq22c2*, which is equal to P - Partially Achieved. This rating was given considering that there exists 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, the team did not consider them 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 third row of Table 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, the assessment team gave a rating equal to P - Partially Achieved, 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 we have given ratings to the atomic requirements, 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.



### 5.2. ArchiMate

The model is presented in Figure 11. The process is similar to the one presented in Figure 10 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.

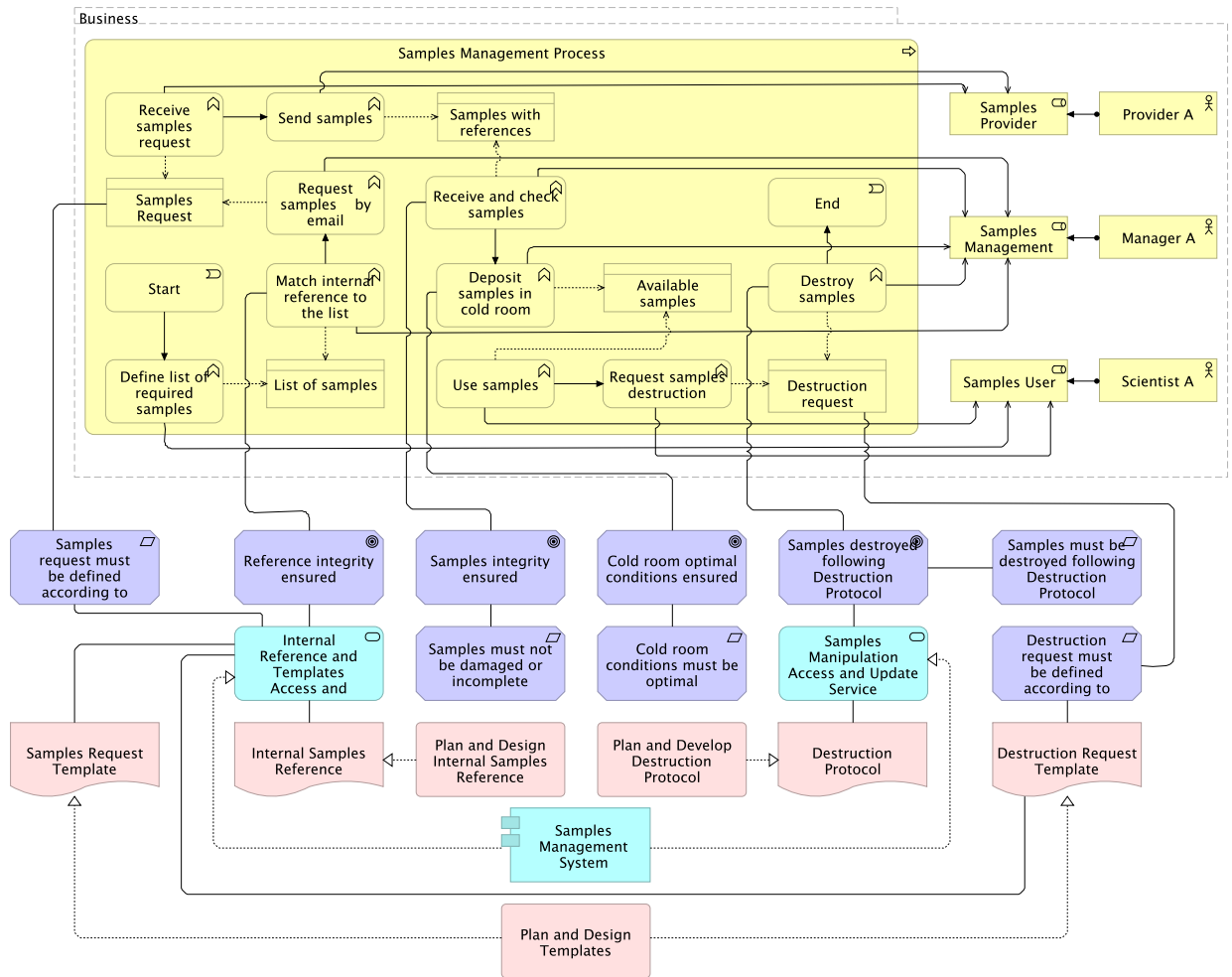


Figure 11: ArchiMate model describing the Samples Management Process.

Similarly to BPMN, as a result of the application of the method presented in Section 4, we already have information regarding the Indicators of ISO/IEC 33020 that can be assessed through enterprise models defined using ArchiMate. We also only consider Indicators from Capability Level 2 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 given by the assessors for the Indicators of the PAs are presented in Figure 12.

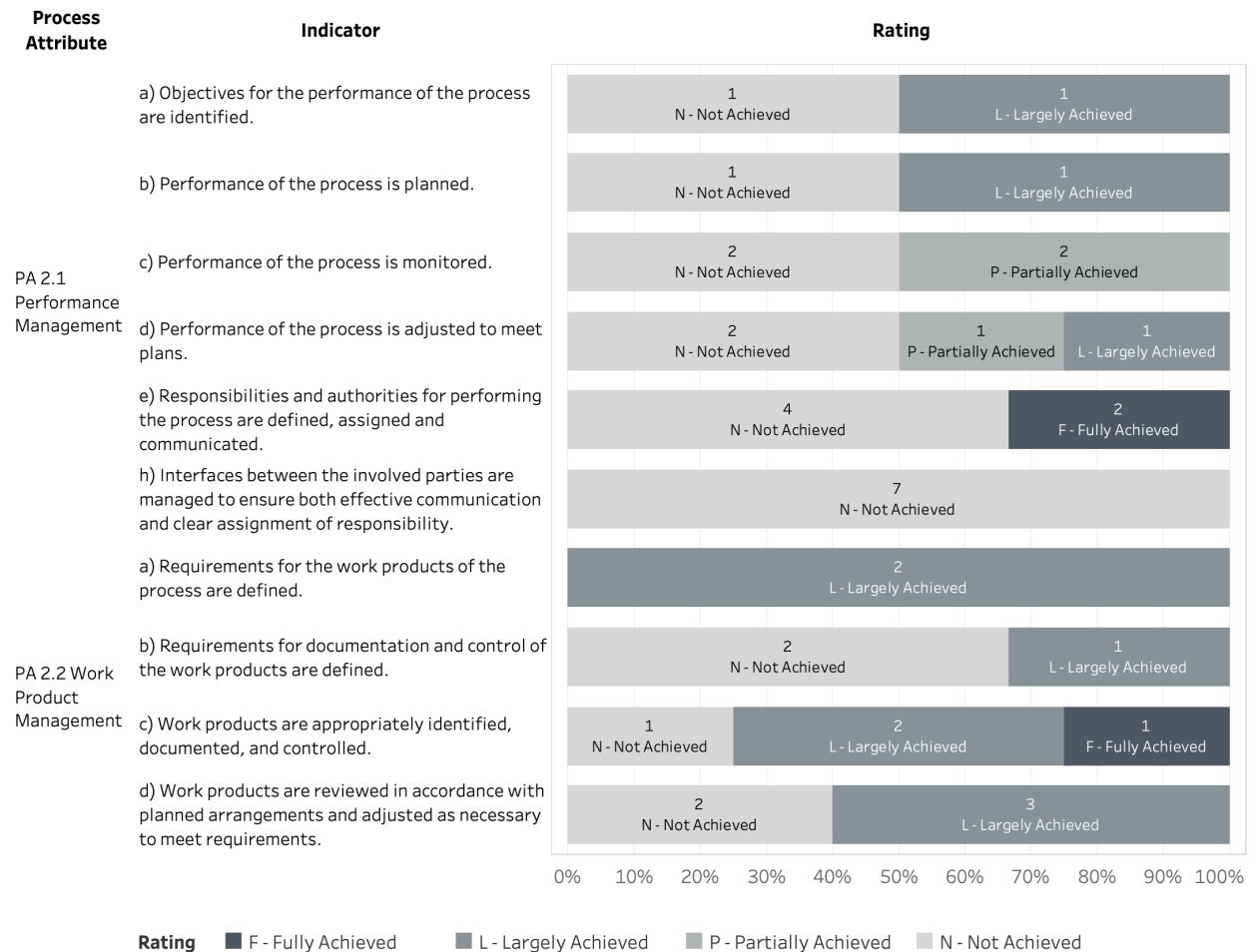


Figure 12: Ratings given by the assessors to the Indicators of the Process Attributes 2.1 and 2.2 of the ISO/IEC 33020 for the Samples Management Process, considering the ArchiMate model.

Results for PA 2.1 show several N - Not Achieved ratings for different atomic requirements. Specially for the requirements extracted from the Indicator 2.2.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, as it was concluded by the assessment team. 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 an 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 EOs. The functions that have links to plans or objectives are: *Match internal reference to list*; *Receive and check samples*; and *Destroy samples*. The assessors concluded that 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 assessors considered that the responsibilities of the process are 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, we have aggregated the ratings given to the atomic requirements to obtain the final rating for each Indicator. We followed the same aggregation method that has been used in the previous section. The final results for the Indicators of PAs 2.1 and 2.2 are shown in Table 8.

Indicator	Rating
2.1.a. Objectives for the performance of the process are identified. b) Performance of the process is planned.	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 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.

### 5.3. 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 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. This will be explored in future work. The case study also confirmed what has been concluded in the 4 section: 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 only. 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.

## 6. Conclusion

This work introduced a method to analyse the suitability of 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, 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. We consider that the proposed approach will allow assessors to understand (*a priori*) which requirements from an assessment framework can be analysed using the enterprise models that exist in the organisation. Moreover, the method can also be applied during proper assessments, in which concrete enterprise models are analysed to define the conformance of the assessed entity to some requirement.

To test the suitability of the proposed approach, we performed a method validation focused on evaluating the suitability of the BPMN and ArchiMate modelling languages to answer requirements defined by the ISO/IEC 33020 international standard, which is 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. We also performed 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. 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 we already had information regarding which requirements are able to be answered with these modelling languages (defined through the method validation), we only focused on concrete Indicators that were analysable through each language, thus reducing the effort necessary to perform the assessment.

Future work will focus on testing the capability of the proposed approach to tackle the combination of models stemming from various modelling languages. Moreover, 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. Indeed, 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. Moreover, efforts will be directed

towards the development of smart systems able to consume enterprise models and process them automatically. Finally, 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, this could also be further investigated in future work.

## References

- [1] H. A. Proper, Enterprise architecture: informed steering of enterprises in motion, in: International Conference on Enterprise Information Systems, Springer, 2013, pp. 16–34.
- [2] G. da Silva Serapião Leal, W. Guédria, H. Panetto, An ontology for interoperability assessment: A systemic approach, Journal of Industrial Information Integration 16 (2019) 100100. doi:<https://doi.org/10.1016/j.jii.2019.07.001>.
- [3] O. Ozcan-Top, O. Demirörs, A reference model for software agility assessment: AgilityMod, in: International Conference on Software Process Improvement and Capability Determination, Springer, 2015, pp. 145–158.
- [4] A. Van Looy, A. Shafagatova, Business process performance measurement: a structured literature review of indicators, measures and metrics, SpringerPlus 5 (1) (2016) 1797.
- [5] T. De Bruin, R. Freeze, U. Kaulkarni, M. Rosemann, Understanding the main phases of developing a maturity assessment model, ACIS Proceedings (2005).
- [6] ISO, 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).
- [7] A. Van Looy, M. De Backer, G. Poels, Defining business process maturity: a journey towards excellence, Total Quality Management & Business Excellence 22 (11) (2011) 1119–1137. arXiv:<https://doi.org/10.1080/14783363.2011.624779>, doi:10.1080/14783363.2011.624779.
- [8] ISO, ISO/IEC 33002:2015 Information technology — Process assessment — Requirements for performing process assessment, Standard, International Organization for Standardization, Geneva, CH (mar 2015).
- [9] M. Gruninger, M. S. Fox, The logic of enterprise modelling, in: Modelling and Methodologies for Enterprise Integration, Springer, 1996, pp. 140–157.
- [10] F. Vernadat, Enterprise modelling: Research review and outlook, Computers in Industry 122 (2020) 103265.
- [11] OMG, Business Process Model and Notation (BPMN), Version 2.0 (January 2011).
- [12] I. Band, T. Ellefsen, B. Estrem, M. Iacob, H. Jonkers, M. Lankhorst, D. Nilsen, H. Proper, D. Quartel, S. Thorn, Archimate 3.0 specification, The Open Group (2016) 978–9401800471.
- [13] Cambridge University Press, Cambridge online dictionary, Cambridge University Press, 2008.
- [14] G. d. S. S. Leal, W. Guédria, H. Panetto, Interoperability assessment: A systematic literature review, Computers in Industry 106 (2019) 111–132.
- [15] W. Guédria, Y. Naudet, D. Chen, Maturity model for enterprise interoperability, Enterprise Information Systems 9 (1) (2015) 1–28. doi:10.1080/17517575.2013.805246.
- [16] L. Von Bertalanffy, The history and status of general systems theory, Academy of management journal 15 (4) (1972) 407–426.
- [17] M. Kohlegger, R. Maier, S. Thalman, Understanding Maturity Models Results of a Structured Content Analysis, 2009.
- [18] J. B. S. D. Santos-Neto, A. P. C. S. Costa, Enterprise maturity models: a systematic literature review, Enterprise Information Systems 13 (5) (2019) 719–769. doi:10.1080/17517575.2019.1575986.
- [19] M. Romero, W. Guédria, H. Panetto, B. Béatrix, Towards Smart Assessment: A Metamodel Proposal, in: 15th OTM/IFAC/IFIP International Workshop on Enterprise Integration, Interoperability and Networking (EI2N 2019), Springer, 2019, pp. 12–22.
- [20] 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).
- [21] D. Karagiannis, H. Kühn, Metamodeling platforms, in: EC-Web, Vol. 2455, 2002, p. 182.
- [22] I.-I. T. Force, Geram: Generalised enterprise reference architecture and methodology, IFIP-IFAC Task Force on Architectures for Enterprise Integration March Version 1 (3) (1999).
- [23] P. Lara, M. Sánchez, J. Villalobos, Enterprise modeling and operational technologies (ot) application in the oil and gas industry, Journal of Industrial Information Integration 19 (2020) 100160. doi:<https://doi.org/10.1016/j.jii.2020.100160>.
- [24] B. Vallespir, Y. Ducq, Enterprise modelling: from early languages to models transformation, International Journal of Production Research 56 (8) (2018) 2878–2896.
- [25] P. Bernus, L. Nemes, R. Morris, The meaning of an enterprise model, in: Modelling and methodologies for Enterprise Integration, Springer, 1996, pp. 183–200.
- [26] E. Trunzer, A. Wullenweber, B. Vogel-Heuser, Graphical modeling notation for data collection and analysis architectures in cyber-physical systems of systems, Journal of Industrial Information Integration 19 (2020) 100155. doi:<https://doi.org/10.1016/j.jii.2020.100155>.
- [27] ISO, 19439, 2006-enterprise integration-framework for enterprise modelling, International Standards Organization, Geneva, Switzerland (2006).
- [28] P. Bernus, Some thoughts on enterprise modelling, Production Planning & Control 12 (2) (2001) 110–118.
- [29] S. H. Lim, N. Juster, A. de Pennington, Enterprise modelling and integration: a taxonomy of seven key aspects, Computers in Industry 34 (3) (1997) 339–359.
- [30] F. Vernadat, Enterprise modeling in the context of enterprise engineering: State of the art and outlook, International Journal of Production Management and Engineering 2 (2) (2014) 57–73.

- [31] J. Rumbaugh, I. Jacobson, G. Booch, Unified modeling language reference manual, the, Pearson Higher Education, 2004.
- [32] C. Menzel, R. J. Mayer, The idel family of languages, in: Handbook on architectures of information systems, Springer, 1998, pp. 209–241.
- [33] A.-W. Scheer, Architecture of integrated information systems (aris), in: Business Process Engineering, Springer, 1994, pp. 4–16.
- [34] G. Keller, A.-W. Scheer, M. Nüttgens, Semantische Prozeßmodellierung auf der Grundlage" Ereignisgesteuerter Prozeßketten (EPK)", Inst. für Wirtschaftsinformatik, 1992.
- [35] F. Vernadat, Ueml: towards a unified enterprise modelling language, International Journal of Production Research 40 (17) (2002) 4309–4321.
- [36] V. Anaya, G. Berio, M. Harzallah, P. Heymans, R. Matulevicius, A. Opdahl, H. Panetto, M. J. Verdecho, B. Vallespir, Dem4: Tutorial on ueml (2007).
- [37] G. Doumeingts, méthode de conception des systèmes en productique, Automatic Control, Université de Bordeaux I (1984).
- [38] D. Chen, G. Doumeingts, The grai-gim reference model, architecture and methodology, in: Architectures for Enterprise Integration, Springer, 1996, pp. 102–126.
- [39] E. C. AMICE, Management overview, CIMOSA: Open System Architecture for CIM (1993) 9–29.
- [40] F. Vernadat, The cimosa languages, in: Handbook on architectures of information systems, Springer, 1998, pp. 243–263.
- [41] F. Vernadat, Cimosa: 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, 1993, pp. 65–84.
- [42] T. O. Group, Business process model and notation 2.0 specification, The Open Group (2011).
- [43] S. A. White, Introduction to bpmn, Ibm Cooperation 2 (0) (2004) 0.
- [44] A. Josey, M. Lankhorst, I. Band, H. Jonkers, D. Quartel, An introduction to the archimate® 3.0 specification, White Paper from The Open Group (2016).
- [45] D. Proença, J. Borbinha, Information governance maturity assessment using enterprise architecture model analysis and description logics, in: International Conference on Theory and Practice of Digital Libraries, Springer, 2019, pp. 265–279.
- [46] S. I. Besik, J.-C. Freytag, Ontology-based privacy compliance checking for clinical workflows (2019).
- [47] O. M. Kherbouche, A. Ahmad, M. Bouneffa, H. Basson, Ontology-based change impact assessment in dynamic business processes, in: 2013 11th International Conference on Frontiers of Information Technology, IEEE, 2013, pp. 235–240.
- [48] C. Ghidini, M. Rospocher, L. Serafini, A formalisation of bpmn in description logics, FBK-irst, Tech. Rep. TR (2008) 06–004.
- [49] R. Almeida, R. Lourinho, M. M. da Silva, R. Pereira, A model for assessing cobit 5 and iso 27001 simultaneously, in: 2018 IEEE 20th Conference on Business Informatics (CBI), Vol. 1, IEEE, 2018, pp. 60–69.
- [50] ISO, ISO/IEC 27001: Information Security Management, Standard, International Organization for Standardization, Geneva, CH (mar 2013).
- [51] A. Lięża, T. Potempa, Artificial intelligence for knowledge management with bpmn and rules, in: IFIP International Workshop on Artificial Intelligence for Knowledge Management, Springer, 2012, pp. 19–37.
- [52] J. Samalikova, R. J. Kusters, J. J. Trienekens, A. Weijters, Process mining support for capability maturity model integration-based software process assessment, in principle and in practice, Journal of Software: Evolution and Process 26 (7) (2014) 714–728.
- [53] W. Van Der Aalst, Process mining: discovery, conformance and enhancement of business processes, Vol. 2, Springer, 2011.
- [54] A. Awadid, D. Bork, S. Nurcan, Towards assessing the multi-view modeling capability of enterprise modeling methods, in: IFIP Working Conference on The Practice of Enterprise Modeling, Springer, 2018, pp. 351–361.
- [55] F. Peres, J. Yang, M. Ghazel, A formal framework for the formalization of informal requirements, The International Journal of Soft Computing and Software Engineering 2 (8) (2012) 14–27.
- [56] V. R. Basili, Applying the goal/question/metric paradigm in the experience factory, Software Quality Assurance and Measurement: A Worldwide Perspective 7 (4) (1993) 21–44.
- [57] C. Kleissner, Enterprise objects framework: a second generation object-relational enabler, in: Proceedings of the 1995 ACM SIGMOD international conference on Management of data, 1995, pp. 455–459.
- [58] G. S. Leal, W. Guédria, H. Panetto, Enterprise interoperability assessment: a requirements engineering approach, International Journal of Computer Integrated Manufacturing (2020) 1–22.
- [59] R. Lalitha, B. Latha, 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 6 (10) (2016) 2114–2127.
- [60] E. A. Emerson, J. Y. Halpern, “sometimes” and “not never” revisited: on branching versus linear time temporal logic, Journal of the ACM (JACM) 33 (1) (1986) 151–178.
- [61] E. M. Clarke, E. A. Emerson, Design and synthesis of synchronization skeletons using branching time temporal logic, in: Workshop on Logic of Programs, Springer, 1981, pp. 52–71.
- [62] ISO, ISO/IEC 33001:2015 Information technology — Process assessment — Concepts and terminology, Standard, International Organization for Standardization, Geneva, CH (mar 2015).