Using Ontologies for Enterprise Architecture and COBIT 5 Analysis

Ana Margarida Fernandes Silva

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Dr. Miguel Leitão Bignolas Mira da Silva
Prof. Dr. Artur Miguel Pereira Alves Caetano

Examination Committee

Chairperson: Prof. Dr. João António Madeiras Pereira
Supervisor: Prof. Dr. Miguel Leitão Bignolas Mira da Silva
Member of the Committee: Prof. Dr. José Luis Brinquete Borbinha

November 2016

Acknowledgments

First and foremost, I would like to thank prof. Miguel Mira da Silva, for his constant support and motivation. Also, for his inspiring, innovating and pragmatic orientation, and above all, for his constant availability and enlightening conversations.

A special thanks to my colleague Nuno Teles for his support and availability in shared parts of this work. Also, I would like to thank Rafael Almeida for his availability to discuss ideas and review this research work.

A very special thanks is due to my parents, Alberto and Juvelina, for giving me the opportunity to study and grow as a person. Thank for all your support and patience throughout this seven years. Thank you for teaching me to believe that hard work always pays off, for teaching me to be persevering, above all, for teaching me to be ambitious while respectful. Thank you also to my sister Joana, for always believing in me and for always giving me the strength to never give up.

To my friends Patricia, André and Paulinho for all these years of friendship, for your support, for all the funny moments, for always being there for me in times of need.

And last but not least, a huge thank you to my boyfriend Paulo, my companion, my best friend, my lover. Thank you for sharing this adventure with me, thank you for all your support, for all your kind words in times of need. Thank you for all the funny moments, for all the nights spent working together, for always believing in me. Thank you for never giving me up, for never letting me down.

I dedicate this thesis to my grandmother Elisa, I am sure you would certainly be more than proud of my achievement.

Abstract

COBIT 5 recommends the Enterprise Architecture (EA) practice as a support to guide the creation and maintenance of its governance and management enablers. These enablers are a set of components that, individually or collectively, influence the success of governance and management over enterprise IT. In particular, process capability assessment is acknowledged as an arguably condition for a successful ITG implementation.

The EA practice delivers the analysis and planning support that is indispensable for effective ITG, and in which models play a central role to fulfill such activities. EA model analysis supports the assessment, optimization and adaptation of organizational systems. Hence, EA model analysis can be used to support COBIT 5 process assessments.

The analysis of an EA model to support COBIT 5 process assessment must be able to answer questions regarding such activity. However, typical EA modeling languages, such as Archimate, are built to address general EA concerns. Hence, the Archimate language by itself does not suffice to address COBIT 5 assessment analysis needs.

In order to overcome this issue we propose an ontology-based approach to extend the Archimate's meta-model with the COBIT 5 concepts and properties required to address the earlier referred analysis needs, and to automatically analyze models resulting from the instantiation of the resulting ontology.

Such proposal was demonstrated by applying the solution artifact to the Archisurance case study. This research work was guided by DSR principles and guidelines, and according to which we have evaluated the proposed solution artifact's efficacy and utility.

Keywords: COBIT 5 PRM, COBIT 5 PAM, Archimate, Ontology, OWL, SPARQL

Resumo

A ferramenta COBIT 5 recomenda a prática de Arquitectura Empresarial (AE) como suporte à criação e manutenção dos facilitadores de gestão e governação das Tecnologias de Informação (GTI). Estes facilitadores influenciam o sucesso da utlização de TI na geração de valor de negócio para as partes interessadas. Em particular, a avaliação de capacidade de processos COBIT 5 é reconhecida como uma condição indiscutivelmente necessária para uma implementação de GTI bem sucedida.

A prática de AE suporta uma GTI informada, fornecendo a análise necessária para tal actividade. A análise de modelos de AE permite a avaliação e optimização de elementos arquitecturais, podendo ser usada como suporte à avaliação de processos COBIT 5. Assim, deve ser capaz de responder a um conjunto de perguntas relacionadas com tal actividade.

No entanto, as linguagens tipicas de modelação de AE, como por exemplo o Archimate, são construidas para satisfazer propósitos gerais de AE. Desta forma, a linguagem Archimate por si só não é capaz de satisfazer as necessidades de análise que concernem a avaliação de processos COBIT 5.

Assim, propomos uma abordagem baseada na aplicação de ontologias para a extensão do metamodelo Archimate com os conceitos e propriedades necessários para suportar a avaliação de processos, bem como para a análise automática de modelos instanciados de acordo com a ontologia resultante.

O artefacto de solução foi demonstrado através do caso de estudo Archisurance. A investigação foi guiada pelos principios e diretrizes da DSRM, e a eficácia e utilidade do artefacto de solução foram avaliadas de acordo com a metodologia.

Palavras-chave: COBIT 5 MRP, COBIT 5 MAP, Archimate, Ontologia, OWL, SPARQL

Contents

List of Tables

List of Figures

Acronyms

COBIT Control Objectives for Information and Related Technologies.

DSR Design Science Research.

DSRM Design Science Research Methodology.

EA Enterprise Architecture.

IS Information Systems.

IT Information Technology.

ITG IT Governance.

OWL Web Ontology Language.

PAM Process Assessment Model.

PRM Process Reference Model.

RDF Resource Description Framework.

SPARQL Simple Protocol and RDF Query Language.

W3C World Wide Web Consortium.

Chapter 1

Introduction

Information is a key resource for enterprises [ISACA, 2012a], from its creation to its destruction, technology plays a significant role. Information Technology (IT) has increasingly become pervasive in enterprises and business environments. As a result, enterprises and their executives strive to generate business value through IT-enabled investments [ISACA, 2012a], meaning that business benefits should be realized by effective and innovative use of IT.

IT has become crucial in the support, sustainability, and growth of enterprises [De Haes et al., 2013]. However, many enterprises fail to demonstrate concrete and measurable business value obtained from their IT-enabled investments [ISACA, 2012c, De Haes and Van Grembergen, 2004, De Haes et al., 2013, Niemann, 2006]. There is a need to govern and manage IT, so that IT and business strategies are aligned and business value is fulfilled [De Haes and Van Grembergen, 2004, De Haes et al., 2013]. Such necessity is attained by IT Governance (ITG): the leadership and organizational structures and processes that ensure that the organization's IT sustains and extends the organization's strategy and objectives [De Haes and Van Grembergen, 2004].

1.1 Research Motivation

Enterprises increasingly recognize the practical relevance of good-practice frameworks as means to guide the implementation of ITG, maximizing the value of IT investments while reducing its inherent risk [De Haes et al., 2013, Lankhorst, 2009, Niemann, 2006]. Control Objectives for Information and Related Technologies (COBIT) 5 fulfills such purpose, consisting of a good-practice framework that assists enterprises in achieving their objectives for the governance and management of enterprise IT [ISACA, 2012a].

COBIT 5 advocates that efficient and effective ITG initiatives require a holistic approach [ISACA, 2012a, ISACA, 2012c]. This approach takes into account a set of interacting components, denominated as "Enablers", which individually or collectively influence the success of governance and management over enterprise IT [De Haes et al., 2013, ISACA, 2012a]. COBIT 5 provides seven categories of enablers, but "Processes" is the one enabler upon which detailed and extensive guidance is described [ISACA, 2012a, ISACA, 2012b]. Moreover, process capability has been a core component of COBIT for more than a decade, and assessing process capability is arguably a necessary condition for the successful implementation of ITG [De Haes et al., 2013, ISACA, 2012a, ISACA, 2012b, ISACA, 2013, ISACA, 2012c].

COBIT 5 recommends the Enterprise Architecture (EA) practice as a support to guide the creation and maintenance of its enablers, more precisely, it advises the implementation of one of its processes "APO03 - Manage Enterprise Architecture" to fulfill such purpose [ISACA, 2012c]. As well as ITG needs good-practice frameworks, such as COBIT 5, to guide the implementation initiatives to maximize the IT investments, it also requires the necessary information for an effective decision making process [Niemann, 2006]. This is achieved by EA: a model-based practice for conducting the analysis, design, planning and implementation of an enterprise's organizational structure, business processes, information systems, and infrastructure [Lankhorst, 2009, Johnson et al., 2004].

Well defined architectures provide an information basis that is crucial for an effective ITG implementation, and act as a complement to the COBIT 5 framework in the context of an ITG initiative [Lankhorst, 2009, Niemann, 2006]. There is a vital connection between EA and ITG, as EA delivers the analysis and planning support that is indispensable for effective ITG [Niemann, 2006]. The analysis of models resulting from EA practices support the assessment, optimization and adaptation of organizational systems. Therefore, EA model analysis can be used to support COBIT 5 process assessments, as it provides the required information to perform such activity.

1.2 Research Problem

COBIT 5 process assessment provides enterprises with a systematic practice to determine the level of capability of their governance and management of IT processes. Determining the level of process capability for a given process enables organizations to perform high level "as-is" and "to-be" health checks to support the governance and management body with relevant information for an informed decision making with regard to process improvement. Such assessments are based on the COBIT 5 Process Assessment Model (PAM), which in turn is based on COBIT 5 Process Reference Model (PRM) [ISACA, 2012a, ISACA, 2012b, ISACA, 2013, ISACA, 2012c].

COBIT 5 PAM is composed of a set of process performance and capability indicators, which are used to support objective evidence collection that enables an assessor to assign ratings to a given process [ISACA, 2013]. The evidence collection is performed by retrieving answers to a set of questions concerned with enterprise artifacts, which can be found on a EA model. Such questions rely on the aforementioned indicators, which, among others, include Practices, Work Products and Process Goals to measure its achievement.

Archimate is the standard [Standard and Group, 2013] developed by The Open Group, consisting of a graphical modeling language to describe EAs. It offers an integrated architectural approach that describes and visualizes the different architecture domains and their underlying relations and dependencies [Standard and Group, 2013, Lankhorst, 2009, Lankhorst and van Drunen, 2007]. Archimate is a high-level modeling language in which the different conceptual domains and their relations are described at a sufficiently abstract level, facilitating a service-oriented and model-driven approach to EA [Lankhorst, 2009, Lankhorst and van Drunen, 2007].

EA model analysis is an activity in which enterprise models are examined in order to extract meaningful information to support the assessment, optimization and adaptation of organizational systems, providing the necessary information for an informed decision making process [Lankhorst, 2009, Johnson et al., 2004, Niemann, 2006]. To perform an effective analysis, the models that serve as basis for the information extraction must fulfill the analysis needs. A model is built to fulfill a purpose; when building a model, its imperative characteristic is to be able to answer a set of questions that address its stakeholders concerns [Lankhorst, 2009, Johnson et al., 2004]. Simply put, in this particular case, if an EA model is analyzed with the purpose of supporting evidence collection for COBIT 5 process assessment, it must be able to answer a set of questions concerned with such activity.

For instance, when determining if a given process achieves capability level 1, according to COBIT 5 PAM, an assessor has to collect evidence that the purpose of the process is fulfilled [ISACA, 2013]. This requires answers to questions such as *"Is there any Work Product not produced by the given process?"* or *"Is there any Process Goal not fulfilled?"*. Thus, if an EA model serves the purpose of supporting the evidence collection for COBIT 5 process assessment, such a model must be able to answer questions as the example ones mentioned above.

These questions are answered by constructing viewpoints that generate views over the EA models [Lankhorst, 2009], in order to address COBIT 5 assessment related concerns. To fulfill this purpose, viewpoints must contain COBIT 5 domain related concepts and relations, which are not to be found in a domain independent modeling language such as Archimate, which serves a general EA modeling purpose. More precisely, COBIT 5 concepts such as *Practice*, *Work Product* and *Process Goal* that are not part of the Archimate meta-model.

Moreover, EA models size and level of detail and complexity can make its analysis exclusively by human means a hard task [Lankhorst, 2009]. Also, as Archimate is a graphical language, it is not prone to automatic analysis, as well as COBIT 5, a set of documents written in natural language, which is also a large and complex set of guidance [De Haes et al., 2013]. Therefore, the problem that this thesis aims to solve can be defined as:

The analysis of EA representations, modeled solely with the Archimate language, is not able to systematically and unequivocally answer a set of questions that support the evidence collection for COBIT 5 Process Assessment.

Hence, we establish the solution objective as :

To provide an ontology-based systematic approach capable of answering questions that support evidence collection for COBIT 5 process capability level 1 assessment, based on the automatic analysis of Archimate models enhanced with COBIT 5 concepts and properties.

Our research proposal is, by means of ontology representation and integration extend the Archimate's meta-model with the COBIT 5 concepts and properties pertaining to the process assessment domain, addressing the earlier referred analysis needs. As well as, to explore the use of ontologies to the automatic analysis of models resulting from the instantiation of the extended Archimate's meta-model. This approach shall be further detailed and described in chapter 3.

1.3 Research Methodology

This research work was undertaken, based on principles and guidelines of Design Science Research (DSR) in Information Systems (IS) [Hevner et al., 2004]. DSR is fundamentally a problem solving paradigm. Its main objective is to create and evaluate IT artifacts intended to solve identified organizational problems. The design process encompasses the creation of innovative products, as well as its evaluation, which in turn provides feedback information and a better understanding of the problem in order to improve both the quality of the product and the design process [Hevner et al., 2004]. Such activities are typically iterated a number of times before the final design artifact is generated [Hevner et al., 2004].

The development of the artifact should be a search process that draws from existing theories and knowledge to come up with a solution to a defined problem [Peffers et al., 2007]. The utility, quality and efficacy of artifacts on solving the identified problem should be assessed [Hevner et al., 2004]. These IT artifacts can be **Constructs** (vocabulary and symbols), **Models** (abstractions and representations), **Methods** (algorithms and practices), and **Instantiations** (implemented prototype systems) [Hevner et al., 2004].

Peffers et al. proposed a methodology to carry out DSR in IS, by providing an iterative process model composed of six activities [Peffers et al., 2007]. These activities are:

1. <u>Problem Identification and Motivation:</u> Definition of the specific research problem and justification of the value of its solution.
2. <u>Define the objectives for a solution:</u> Extraction of the objectives of the solution from the problem definition and verification of its feasibility.
3. <u>Design and Development:</u> Creation of the artifact, involving the necessary activities to determine the artifact's desired functionality and its architecture.
4. <u>Demonstration:</u> Demonstration of the use of the artifact to solve one or more instances of the problem.
5. <u>Evaluation:</u> Assessment of how well the artifact supports the solution to the defined problem. It involves the comparison of the solution objectives and the actual results obtained from the use of the artifact in the demonstration.
6. <u>Communication:</u> Communication of the problem and its importance, the developed artifact, its utility and novelty, the rigor of its design, and its effectiveness to relevant audience.

Figure 1.1 depicts the Design Science Research Methodology (DSRM) steps applied to our research work.

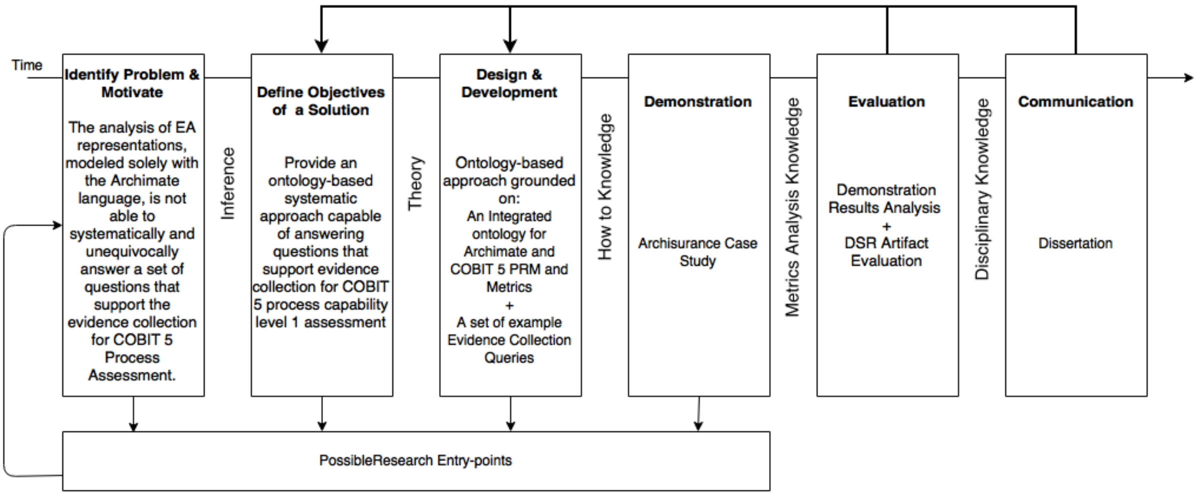

Figure 1.1: DSRM steps applied to our research.

1.4 Dissertation Outline

We now describe how this document is organized:

- **Chapter 1** - This chapter presents the motivation and problem behind this research work.
- **Chapter 2** - This chapter presents the state of the art regarding the research scope and the necessary technologies for the solution proposal.
- **Chapter 3** - This chapter presents the proposed solution artifact and the necessary steps for its design and development.
- **Chapter 4** - This chapter presents the performed utility demonstration of the proposed solution artifact.
- **Chapter 5** - This chapter presents the performed evaluation of the proposed solution artifact.
- **Chapter 6** - This chapter presents an overview of the realized research work, its limitations and future work.

Chapter 2

Background

This chapter supports the DSRM steps of problem identification and motivation, as well as the solution objectives definition. We first introduce the COBIT 5 framework, and give an overview of COBIT 5 PRM and PAM. EA is introduced, and its various categories of analysis presented. The Archimate language is also presented as the standard language for EA modeling, along with a brief description of its meta-model. Ontologies and ontology techniques are described, along with the semantic web technologies that shall support our solution proposal. And finally, we present work already performed towards the ontological representation and integration of Archimate and COBIT 5 domains.

2.1 COBIT 5

COBIT 5 is a good-practice framework that provides enterprises with guidance on how to achieve their objectives for governance and management of enterprise IT. It helps enterprises to create optimal value from IT by maintaining a balance between realizing benefits and optimizing risk levels and resource use [ISACA, 2012a]. COBIT 5 relies on a set of five principles, namely *"1 - Meeting stakeholders needs"*, *"2 - Covering the enterprise end-to-end"*, *"3 - Applying a single integrated framework"*, *"4 - Enabling a holistic approach* and *"5 - Separating governance from management"*.

COBIT 5 advocates that enterprises exist to create value for their stakeholders, and such value creation should be attained through benefits realization at an optimal resource cost while optimizing risk. As such, it provides a valuable tool, the *Goals Cascade*, which translates stakeholder needs into an enterprise's actionable strategy [ISACA, 2012a]. This translation is achieved by mapping stakeholder needs into specific, actionable and customized enterprise goals, IT-related goals and enabler goals.

Principle *"4 - Enabling a holistic approach"* advocates that an effective and efficient ITG initiative requires a holistic approach. According to the COBIT 5 framework, such an approach takes into account a set of enablers - factors that, individually and collectively, influence the success of governance and management over enterprise IT. These enablers are classified according to seven categories, but "Processes" is the one enabler upon which detailed and extensive guidance is described [ISACA, 2012a, ISACA, 2012b].

The enabler *"Processes"* is described in the COBIT 5 PRM. COBIT 5 PRM describes and defines in detail a set of thirty seven processes for governance and management of IT. This model represents all of the processes normally found in an enterprise, related to IT activities, and offers a common reference model understandable to operational IT and business managers. The COBIT 5 PRM is a complete, comprehensive model, but it is not the only possible process model. Each enterprise must define its own process set, taking into account its specific situation [ISACA, 2012a, ISACA, 2012b].

Each process contained in the COBIT 5 PRM is described to fulfill a purpose, which is further detailed in process goals. Information concerning the COBIT 5 goals cascade is also contained in such model, in the form of IT related goals which in turn support enterprise goals and stakeholder needs. For each goal a set of metrics is defined so that its achievement can be measured. Also, each process is composed of a set of practices, further detailed in activities, and which input and output a set of work products. Process goals are supported by work products and practices. In the context of such practices, RACI charts are described so that stakeholders responsibility levels are documented [ISACA, 2012a, ISACA, 2012b, ISACA, 2013].

The assessment of a process, according to the COBIT 5 PAM [ISACA, 2013], enables the determination of the current capability of such process. The process capability is expressed in terms of process attributes grouped into five capability levels, ranging from level *"0 - Incomplete Process"* up to level *"5 - Optimizing process"*. The capability level of a process is determined by verifying the achievement of specific process attributes, based on process *performance* and *capability* indicators [ISACA, 2013].

In particular, the assessment of process capability level *"1 - Performed Process"* relies on the achievement of process *performance attribute*, which in turn is based on the verification of *performance indicators*. Achieving capability level 1 means that the process being assessed fulfills the purpose for which it was designed; the performance indicators include: process goals, practices, and work products [ISACA, 2013]. Hence, according to the COBIT 5 PAM, the verification of capability level 1 achievement can be performed in three ways:

1. Search for evidence that the process outcomes, as they are described for each process, are achieved [ISACA, 2013].
2. Search for evidence that the intent of the base practice is being performed [ISACA, 2013].
3. Verify if work products are produced that provide evidence of process outcomes [ISACA, 2013].

Additionally, the COBIT 5 PRM comes with a set of metrics that are advised by the framework itself as means to manage the performance of the processes enabler. Since, for process capability level 1, the attribute rated is process performance, such metrics can be used to measure the achievement of COBIT 5 process goals, providing sufficient evidence for process performance assessment [ISACA, 2012a].

The remaining levels are assessed by verifying the achievement of their correspondent process capability attributes, based on *capability indicators*: generic practices and generic work products [ISACA, 2013]. The overall COBIT 5 process assessment is an evidence based activity [ISACA, 2013], which relies on set of questions concerning enterprise artifacts.

2.2 Enterprise Architecture

Lankhorst et al. defines EA as "a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure" [Lankhorst, 2009]. EA captures the essentials of the business, IT and its evolution. It offers a holistic perspective of the current and future operations, and on the actions that should be taken to achieve the enterprise's goals, thus facilitating the translation from corporate strategy to daily operations [Lankhorst, 2009].

There is a vital connection between EA and ITG, as EA delivers the analysis and planning support that is indispensable for an effective ITG [Niemann, 2006]. EA represents the CIO's management information system, making the IT portfolio analyzable and supporting the strategic planning process by providing "as-is" and "to-be" models [Niemann, 2006, Johnson et al., 2004]. Authors recognize the relevance of models for the representation of the essence of an EA in an unambiguous form [Lankhorst, 2009, Johnson et al., 2004, Niemann, 2006].

A model is an abstract and unambiguous conception of a domain, built to fulfill a purpose and meant to answer questions [Lankhorst, 2009]. It is rarely the case in which a stakeholder has interest in the full scope and detail of models representing EA. As such views over such models are required to address specific stakeholder's concerns. A *view* is specified by means of a *viewpoint*, which in turn prescribes the conventions for constructing and using a *view*. A *viewpoint* must contain the necessary concepts and relations to address the specific concerns [Lankhorst, 2009, Johnson et al., 2004].

The value of an EA model increases if it can be used to support the decision making process [Lankhorst, 2009]. Enterprise Architecture model analysis is an activity in which enterprise models are examined in order to extract meaningful information to support the assessment, optimization and adaptation of organizational systems [Lankhorst, 2009, Johnson et al., 2004, Niemann, 2006]. Lankhorst et al. classifies EA analysis according to two different techniques: *functional* allowing to understand how a system that conforms to an architecture works, to find the impact of a change on an architecture, or to validate the correctness of an architecture; and *non-functional* (or *quantitative*) providing performance, reliability and cost measures, among others [Lankhorst, 2009]. Work towards the enhancement of EA models with quantitative aspects as means to enable non-functional analysis was already performed [Lankhorst, 2009, Johnson et al., 2004].

Also, Niemann classifies EA analysis according to eight categories: *Dependency*, *Coverage*, *Interfaces*, *Heterogeneity*, *Complexity*, *Conformity*, *Costs*, *Benefits* [Niemann, 2006]. From these we emphasize the *Conformity* analysis, as it provides means to ascertain the compliance of architectural elements against rules, policies, norms, standards, and reference models [Niemann, 2006].

2.3 Archimate

The Archimate standard [Standard and Group, 2013] was developed by The Open Group, and it consists of a graphical modeling language to describe enterprise architectures. It offers an integrated architectural approach that describes and visualizes the different architecture domains and their underlying relations and dependencies [Standard and Group, 2013].

The core of the Archimate language comprises three main types of elements: *Active* structure elements, *Behavior* elements, and *Passive* structure elements. An *Active* structure element is an entity capable of performing behavior, a *Behavior* element is a unit of activity performed by one or more Active structure elements, and a *Passive* structure element is an object on which behavior is performed [Standard and Group, 2013].

Along with these elements, Archimate defines three main layers according to a “Service Orientation” paradigm, in which each layer exposes functionality in the form of a service to the above layer [Lankhorst, 2009, Standard and Group, 2013]. Archimate also provides a set of relationships to model the link of business, application and technology, so that business-IT alignment can be achieved. The generic meta-model for the core concepts of Archimate can be seen in figure 2.1.

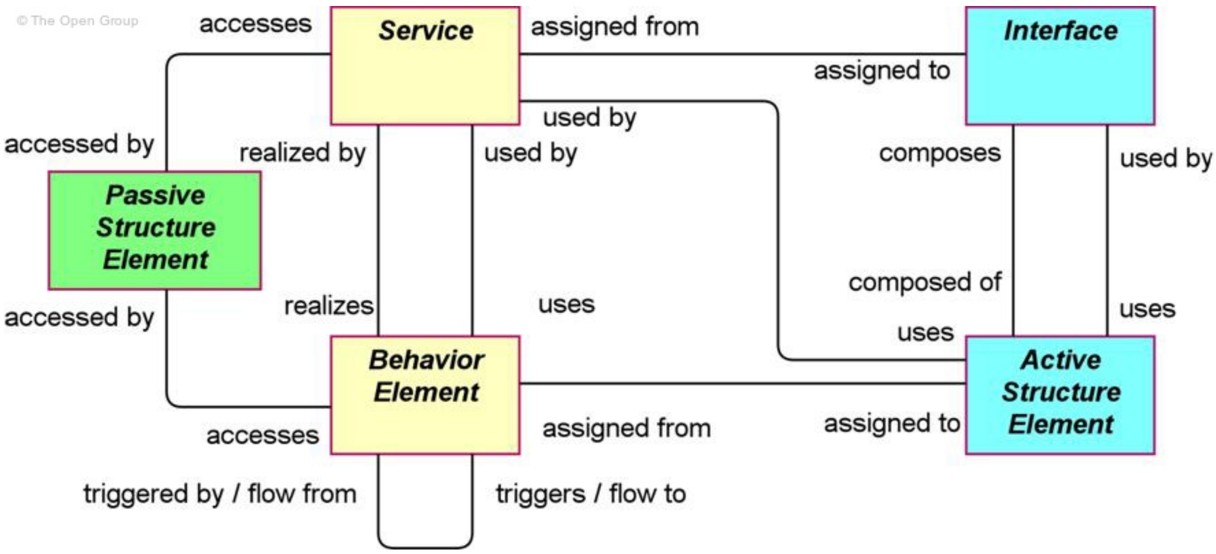

Figure 2.1: Archimate Generic Meta-model. [Standard and Group, 2013]

The three main layers are the *Business* layer, the *Application* layer, and the *Technology* layer. In each of them, the aforementioned core elements are specialized according to each layer nature [Standard and Group, 2013]. The *Business* layer offers products and services to external customers, which are realized in the organization by business processes performed by business actors; the *Application* layer supports the business layer with application services which are realized by (software) applications; and the *Technology* layer offers infrastructure services needed to run applications, realized by computer and communication hardware and system software [Standard and Group, 2013].

The Archimate core language provides the basic concepts and relationships that fulfill the general enterprise architecture modeling needs, however it also provides an extension mechanism which enables stakeholders to address more specific domains. It provides a Motivation extension and an Implementation and Migration extension. The first one (Motivation) addresses the way that the enterprise architecture is aligned with its context, as described by motivational elements, which provide the context or reason lying behind the architecture of an enterprise [Standard and Group, 2013]. The later (Implementation and Migration) adds concepts to support activities such as portfolio and project management.

2.4 Ontologies

The term ontology first appeared in philosophy, as an attempt to classify things in the world, described as a systematic account of Existence [Gruber, 1993, Studer et al., 1998]. Computer Science researchers adopted the term ontology to refer to a computational artifact that formally models the relevant entities and relations of some system of interest [Guarino et al., 2009]. Studer et al. , based on the merging of previous definitions, stated that an ontology, in computer science, is "a formal, explicit specification of a shared conceptualization" [Studer et al., 1998].

A **“conceptualization”** refers to an abstract world view, with respect to a given domain, conceived as a set of concepts, their definitions and the relationships among them [Gruber, 1993, Uschold and Gruninger, 1996]. **“Explicit”** means that the concepts used and the constraints on their relationships should be explicitly defined [Corcho et al., 2006]. **“Formal”** implies that the ontology should be represented in a machine-readable format. And at last, **“shared”** means that the ontology should result from an agreement, capturing consensual knowledge, and be accepted by a group [Corcho et al., 2006].

The main components of an ontology consist of [Corcho et al., 2006]:

- **Classes:** Represent concepts, and which usually are organized in taxonomies.
- **Relations:** Represent associations between concepts of the domain.
- **Axioms:** Represent imposed constraints to the ontology relations and classes, and which are always true.
- **Instances:** Represent elements or individuals of the ontology.

An ontology is a specification used for making ontological commitments: agreements to use a vocabulary in a way that is consistent with respect to the theory (axioms and documentation strings) that specifies the ontology [Gruber, 1993, Studer et al., 1998, Guarino et al., 2009, Uschold and Gruninger, 1996]. Making too many ontological commitments limits extensibility, while making too few can result in the ontology being consistent with incorrect or unintended models [Studer et al., 1998, Uschold and Gruninger, 1996]. Therefore, authors advocate the principle of minimal ontological commitment: specify the weakest theory and define only those terms that are essential to the communication of knowledge consistent with the theory that specifies the ontology [Corcho et al., 2006, Uschold and Gruninger, 1996].

According to Uschold and Gruninger [Uschold and Gruninger, 1996], ontologies facilitate *Communication* between people with different concerns and viewpoints of a certain domain, enhances *Interoperability* enabling different users to exchange data, and supports the design and development of Software Systems providing *Reliability* and *Reusability*.

An ontology can be built from scratch, or it can be constructed by means of existing ontologies integration. *Ontology Integration* has had different meanings in the ontology field. Shortly, it consists of identifying common concepts and relations between two or more ontologies [Pinto et al., 1999]. This process can be subdivided into three techniques [Amrouch and Mostefai, 2012]:

- **Ontology Mapping:** Consists of a process to build a new ontology by describing the semantic relationship between two (or more) concepts from two (or more) ontologies.
- **Ontology Alignment:** Consists of a process to build a new ontology by describing the correspondences between all the equivalent concepts of two ontologies.
- **Ontology Merging:** Consists of a process to build a new ontology by describing the semantic relationship between concepts of several ontologies, and merging them into a single one.

From the application of the previously described techniques, different types of mismatches may arise between different ontologies. These mismatches can be divided into three types [Amrouch and Mostefai, 2012, Noy, 2009]:

- **Syntactic Mismatches:** Consists of syntactic heterogeneity between two ontologies, due to different representation languages usage.

- **Lexical Mismatches:** Consists of heterogeneities between the names of entities, instances, properties or relations of two ontologies. These heterogeneities may take four forms:
  - **Synonyms:** The same entity is represented by two different names.
  - **Homonyms:** The same name represents two different entities.
  - The same name in different languages.
  - The same entities are named with the same words but with different syntactic variations.
- **Semantic Mismatches:** Consists of heterogeneities between the contents of two ontologies, namely between entities, properties and relationships represented in these. These heterogeneities may take three forms:
  - **Coverage:** Two ontologies are different from each other in that they cover different portions (possibly overlapping) of the world (or even of a single domain).
  - **Granularity:** Two ontologies are different from each other in that one provides a more/less detailed description of the same entity.
  - **Perspective:** Two ontologies are different from each other in that one may provide a viewpoint on some domain which is different from the viewpoint adopted in another ontology.

Once an ontology is built and represented in some language, one can reason over it. The term "Reasoning" refers to any mechanism/procedure for making explicit a set of facts that are implicit in an ontology. According to Lenzerini et al. [Lenzerini et al., 2004], there are two main purposes for one to reason over an ontology, namely Validation and Analysis. Performing Validation of an ontology ensures that the ontology is a good representation of the domain of discourse that one is aiming at modeling. By reasoning over an ontology, it is possible to verify if the axioms defined on it are actually being fulfilled. Analysis consists of deducing facts about the domain by reasoning over the ontology, assuming that its representation is a faithful one. It allows, for example, the discovery of new relations between individuals of the ontology.

2.5 Semantic Web

The concept of *Semantic Web* first appeared in 2001 [Berners-Lee et al., 2001], as a way to extend the current web at that time, in which content was mainly designed for humans to read and not for computer programs to manipulate meaningfully. The Semantic Web [Berners-Lee et al., 2001, Shadbolt et al., 2006] combines means for representing structured collections of information, to perform integration of that information through common conceptualizations referred as ontologies, and retrieving that information. Others [DuCharme, 2011] have later defined the Semantic Web as "a set of standards and best practices for sharing data and the semantics of that data over the web for use by applications". These standards are provided by W3C and will be further explained in the remainder of this subsection.

RDF and RDFS

The Resource Description Framework (RDF) is an infrastructure that enables the encoding, exchange and reuse of structured meta-data (data about data). RDF provides meta-data interoperability, by supporting common conventions of semantics, syntax and structure [Miller, 1998].

It provides a simple, yet powerful data model to describe resources [Miller, 1998]. The provided data model consists of three major components: Resources, Properties, and Statements. *Resources* can be any object, as long as they can be uniquely identified by a Uniform Resource Identifier (URI) [Miller, 1998]. *Properties* are specific aspects, characteristics, attributes, or relations used to describe a resource [Klein, 2001]. *Statements* consist of specific resources with named properties plus those properties value for those resources [Klein, 2001].

These Statements are expressed in the form of Triples: **Subject** (resource), **Predicate** (property), **Object** (resource or literal value) [Miller, 1998, Klein, 2001, Berners-Lee et al., 2001]. The list of these triples consists of an RDF description, and can be seen as directed labeled graph.

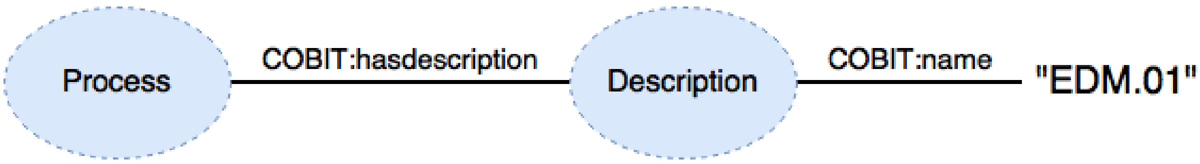

Figure 2.2: Example of a RDF Triple.

RDF is syntax independent, which means an RDF graph can be represented in various syntaxes, namely RDF/XML that was the first one adopted. But others appeared too, such as TURTLE or JSON-LD, among others. These only differ in the way a RDF graph is written, but its triples are the same, and thus are logically equivalent.

Concerning a specific context of application, if one wants to describe meta-data in a particular domain, RDF by itself lacks of a mechanism to define domain-specific properties or resources, thus it is domain-neutral [Klein, 2001]. There is a need to provide a mechanism to define vocabularies so that they can be used in RDF descriptions for a particular context, which is accomplished by **RDF Schema** (RDFS).

RDFS is a simple type system, that enables the definition of class hierarchies representing resources and property hierarchies for those resources, it also allows for the definition of restrictions in the domain or range of a given statement [Klein, 2001]. RDFS is basically an extension of RDF, so its specification uses RDF too. It provides means for representing minimal ontologies, with low expressivity [Shadbolt et al., 2006].

OWL

Depending on the domain of the problem [Shadbolt et al., 2006], one may need greater expressivity in what concerns resources and properties description. The Web Ontology Language (OWL) [Hitzler et al., 2009] makes this kind of more specific and restricted representation possible. OWL's main purpose is to provide an efficient representation of ontologies, supporting various types of inference [Shadbolt et al., 2006]. By extending RDFS, it enriches the possible vocabulary descriptions.

It provides a set of profiles, also called sublanguages, namely OWL Full, OWL DL, OWL EL, OWL QL, and OWL RL. Each of these profiles restricts the power of expressivity and reasoning that one can apply to a given context [DuCharme, 2011]. Also, the OWL language is based on the Open-World Assumption (OWA). OWA is the assumption that what is not known to be true it is simply unknown: this means that if we want to infer some information, relation or dependency, it has to be explicitly specified in the ontology. Otherwise, when reasoning over the ontology to infer that information, it will result in an unknown answer.

SPARQL

RDF graphs are stored in repositories, usually called Triple stores. These repositories vary in their capabilities, and should be chosen according to each one needs [Shadbolt et al., 2006]. The Simple Protocol and RDF Query Language (SPARQL) language is the W3C standard for accessing and querying the RDF graphs stored on those. It was designed to query data conforming to the RDF data model. When performing a SPARQL query, typically one retrieves pieces of information from a certain subset that meet certain conditions [DuCharme, 2011]. These conditions are described using Triple patterns, similar to the RDF ones, but these may contain variables in order to achieve higher flexibility in how they match against the data [DuCharme, 2011].

Jena

Apache Jena[1] (Jena) is an open source framework that enables the development of Semantic Web applications through the writing of Java code. It combines a set of API's that interact together, and each of them provides means to use the standards described in the previous subsections.

It provides an RDF API that enables the development of RDF models, an Ontology API that enables the usage of RDFS and/or OWL to add extra semantics to the RDF data, and a SPARQL API to query the RDF data. It is possible to run SPARQL queries using the ARQ engine, or using the Fuseky SPARQL server over HTTP. Jena also provides means for reading and writing RDF models in various syntaxes, through the usage of parsers and writers. The RDF models can be stored using the Storage API in three possible ways: in memory, SDB using SQL databases, and TDB using Triple stores. Finally, Jena provides an Inference API, in which it is possible to perform reasoning over data and define inference rules.

Protégé

Protégé[2] is an ontology development environment that enables the creation, uploading, modification and sharing of ontologies for collaborative viewing and sharing. It has a highly configurable user interface, which facilitates activities such as ontology integration. It is compliant with the World Wide Web Consortium (W3C) standards, and it has full support for the OWL ontology language. It also supports a number of syntax formats, namely RDF/XML, among others.

2.6 Archimate and Ontologies

Work towards the ontological representation of the Archimate meta-model has already been performed [Bakhshadeh et al., 2014]. The use of ontologies was explored has means to provide a formal computable representation for EA models, in this particular case for Archimate models, enabling the analysis of their consistency and completeness against the represented meta-model. Such representation was performed by extracting all the concepts and relations pertaining to the Archimate language meta-model, and transforming them into an OWL classes and properties . Also, axioms were added to reinforce the meta-model restrictions imposed by the language. An Archimate model for an EA representation can

[1]"Apache Jena". Available at `https://jena.apache.org`. Last accessed on November 4th, 2016.
[2]"Protégé". Available at `http://protege.stanford.edu`. Last accessed on November 4th, 2016.

then be transformed into an OWL ontology, and its elements classified according to the classes and properties contained in the Archimate's meta-model ontology.

It is also acknowledged by other authors that in order to address domain-specific analysis needs for stakeholders, the Archimate language by itself does not suffice [Bakhshandeh et al., 2013, Antunes et al., 2014a, Antunes et al., 2015, Antunes et al., 2014b, Caetano et al., 2016]. To overcome this issue, Antunes et al. explored the federation of multiple models or domain-specific languages [Antunes et al., 2015, Antunes et al., 2014a, Bakhshandeh et al., 2013]. Such federation was proposed by means of ontology integration, more precisely by means of ontology mappings.

A federated schema can have multiple configurations, for instance an upper ontology (UO) can be specified containing the core concepts and relations and being domain-independent in the context of EA, and domain-specific ontologies (DSO), containing the concepts and relations pertaining to a specific domain, can mapped to such ontology via ontology mappings. Or a federated schema can have multiple domain-specific ontologies mapped to each other; either ways the ontology mappings must be traceable to the stakeholder concerns [Antunes et al., 2015, Antunes et al., 2014b, Antunes et al., 2015].

The mentioned federated approach is able to address domain-specific analysis needs, for example Caetano et al. as also performed research towards the specification of a federated schema, in particular for Archimate, Business Model Canvas (BMC) and E3value [Caetano et al., 2016, Caetano et al., 2015]. This schema accompanied with a set of SPARQL queries is able to automatically detect inconsistencies among such models. Antunes et. al has also demonstrated such an approach by representing a domain-specific ontology describing sensors, and by mapping such an ontology to the UO, in this case Archimate ontology, being able to address the specific stakeholders analysis needs, which otherwise would not be possible.

Along this line of research, Antunes et al. has also explored ontology techniques to the several categories described for EA analysis by Niemann [Antunes et al., 2016, Niemann, 2006]. Relying on the schema federation approach, for each type of analysis described, the required properties, not contained in the Archimate's meta-model and required to fulfill the analysis needs, were represented as attributes of Archimate's meta-model elements [Antunes et al., 2016]. To fulfill this, the represented properties were mapped to the Archimates's meta-model ontology elements. Having the integrated ontology, SPARQL queries and inference can be used towards the various types of analysis, providing meaningful information in an automatic way. Note that, such an approach enables the automatic analysis of quantitative aspects of architectural elements, such aspects being represented as properties of such elements.

2.7 COBIT 5 and Ontologies

As for the Archimate language, other authors have already explored the use of ontologies in order to computationally represent the concepts and relations pertaining to the COBIT 5 domain, more precisely to the COBIT 5 PRM. Textor and Geihs [Textor and Geihs, 2015] acknowledge that an informed ITG is in fact more effective and efficient. As such, the authors explored an ontology-based approach for the calculation of metrics, contained in COBIT PRM, as means to attain a better informed IT management, which in turn enables a better IT governance.

Such an approach relies on a set of ontologies: *COBIT 5 PRM*, the already existent ontology *QUDT* used to formalize metrics, as well as ITIL. The COBIT 5 ontology is represented using the OWL language, and whose purpose is to describe processes contained in COBIT 5 PRM, along with the goals supported by such processes and respective metrics to measure its fulfillment. The QUDT ontology, which stands for "Quantity, Unit, Dimension and Type", is used to formalize metrics, meaning that each metric pertaining to the COBIT 5 PRM is further detailed using the concepts composing the QUDT ontology. Hence, metrics are classified according to its dimension, unit and type, and populated with concrete measurement values obtained from IT monitoring tools.

The ITIL ontology is used as means to provide the entities that are linked to metrics, for instance, for a metric regarding the "Number of customer service disruptions due to IT service-related incidents" and formalized with QUDT ontology, elements such as "IT service" and "Incident Records" are mapped to the formalized metric. Moreover, the authors have also mapped COBIT 5 ontology concepts to ITIL concepts. Having such an integrated schema, also a set of SPARQL queries was defined, along with a set of rules relying on reasoning capabilities, so that the resultant populated ontologies could be analyzed and metrics calculated.

As a result from this research work, an OWL representation of COBIT 5 ontology was provided. This ontology contains a set of concepts and relations that belong to an earlier specified COBIT 5 meta-model. Such meta-model was specified based on already existing works towards its definition. Namely, Goeken and Alter [Goeken and Alter, 2009] and Souza Neto and Ferreira Neto [Souza Neto and Ferreira Neto, 2013] who have specified an ontological meta-model for COBIT 4.1, and Moeller et al. [Moeller et al., 2013] who has specified an ontological meta-model for COBIT 5 PRM.

2.8 Archimate and COBIT 5

The integration of Archimate and COBIT 5 domains was already performed, and we now expose the work towards such integration presented by authors Cadete and Almeida et al. [Cadete, 2015, Almeida et al., 2016]. Cadete has proposed to integrate the COBIT 5 process rationale in EA representations, using the standard Achimate extensions in order to promote easy adoption [Cadete, 2015]. The purpose of such integration was to improve the outcomes of COBIT 5 process assessment and process improvement initiatives [Cadete, 2015].

The author built a series of viewpoints, namely a viewpoint for process performance, which shows the main concepts and relationships involved in a process performance assessment, using also concepts from the goals cascade. This viewpoint is built upon an ontological mapping between COBIT 5 process performance assessment concepts and the ArchiMate concepts the author found suitable, based on COBIT 5 documentation, Archimate's standard and the purpose the mappings must fulfill [Cadete, 2015].

These mappings were performed using concepts from both Archimate's Business Layer and the Motivation extension, although with a major focus on the latter. For instance, concepts such as *Practice* and *Work Product* were mapped to Archimate's meta-model concepts from the Motivation extension, in order to allow for a more explicit representation of the entities and relations that represent capability evidence, capability motivations, and their corresponding relations [Cadete, 2015].

Also Almeida et al. has mapped COBIT 5 to Archimate concepts, in this particular case as means to map, model and integrate both COBIT 5 and COSO frameworks, providing a visual representation to help organizations better understand, implement and assess these frameworks simultaneously [Almeida et al., 2016]. Note that an ontological mapping between COSO and Archimate was also specified, but was not considered in this research since it is out of the research scope. The author exposed the need of using a common language and a standardized approach in support of such integration, hence providing a uniform representation and creating a common frame of reference [Almeida et al., 2016].

Following this, Almeida et al. proposed an ontological mapping between COBIT 5 and Archimate meta-models, with a special focus on COBIT 5 PRM and using Archimate concepts also from both Business layer and the Motivation aspect. These mappings were based on Cadete work, and based on the author's interpretation of COBIT 5 documentation and Archimate's specification [Almeida et al., 2016].

Differences between Almeida et al. and Cadete's work rely on the Archimate concepts mapped to COBIT 5 concepts such as *Practice* and *Work Product*. Almeida et al. has mapped these performance indicators to concepts pertaining to the Archimate's Business Layer, while Cadete has mapped them to concepts pertaining to the Archimate's Motivation extension. The chosen ontological mapping shall be further discussed in section 3.

Chapter 3

Research Proposal

This section maps to the DSRM step of *Design and Development*, and in which we present our solution proposal. We now recall the problem this thesis aims to solve:

The analysis of EA representations, modeled solely with the Archimate language, is not able to systematically and unequivocally answer a set of questions that support the evidence collection for COBIT 5 Process Assessment.

Based on the literature review, and on the acknowledged problem, we established our solution objective as:

To provide an ontology-based systematic approach capable of answering questions that support evidence collection for COBIT 5 process capability level 1 assessment, based on the automatic analysis of Archimate models enhanced with COBIT 5 properties.

As mentioned in subsection 2.1, the evidence collection for COBIT 5 process capability level 1 can be subdivided into three possible approaches: (1) *search for evidence that the intent of the process base practices is being performed*, (2) *verify if process work products are produced that provide evidence of process outcomes*, and additionally (3) *search for evidence that process goals are achieved based on metrics*. Following this, and as we aim to use EA models conformity analysis as means to answer questions to support such approaches, we now explain how such analysis shall be performed.

In order to fulfill such approaches we propose:

a) For (1) and (2), the *search for architectural elements, represented on the organization's Archimate model, that realize COBIT 5 practices and work products*.

b) For (3), the *search for non-compliant architectural elements, represented on the organization's Archimate model, according to a set of COBIT 5 metrics*.

As acknowledged in the research problem section, Archimate models by them selves are not able to address such specific analysis needs. As such, we explore the extension of Archimate's meta-model with a set of COBIT 5 concepts and properties, in order to address the required analysis specificity. Following this, to fulfill a) we propose the extension of Archimate's meta-model with COBIT 5 concepts pertaining to COBIT 5 PRM, so that elements contained in Archimate's meta-model can be related to such concepts. Whereas, to fulfill b) we decompose a subset of COBIT 5 metrics into a set of properties, and extend the Archimate's meta-model with such properties, meant to be represented as attributes of its architectural elements.

Therefore, based on the established solution objective and on the above described approaches, the following requirements are raised:

- **R1** - The solution must represent Archimate, COBIT 5 PRM and Metrics ontologies using the OWL language.
- **R2** - The solution must provide means to integrate the Archimate ontology with COBIT 5 PRM and Metrics ontologies.
- **R3** - The solution must provide means to automatically analyze the integrated ontology based on SPARQL queries, providing answers to questions that support evidence collection for COBIT 5 process capability level 1 assessment.

In order to attain the defined solution objective and requirements, we propose:

An approach based on the application of a set of ontology-based techniques to represent and integrate Archimate's meta-model with COBIT 5 concepts and properties, and also to automatically analyze the resulting integrated ontology.

The figure 3.1 gives an overview of our proposed approach. In the next three sections we explain

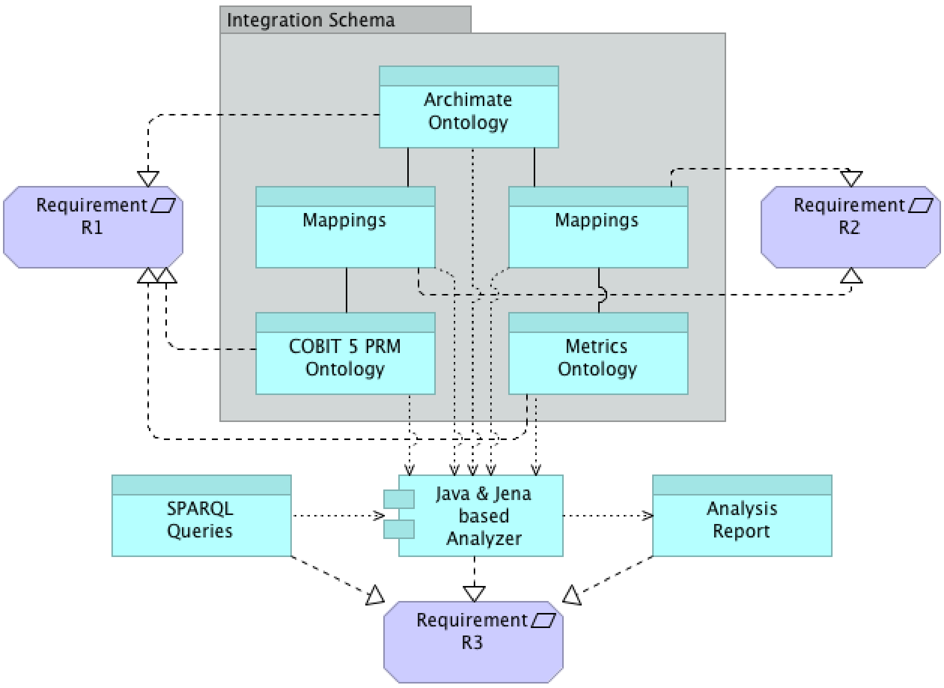

Figure 3.1: An overview of the proposed solution.

how we have realized each of the above defined solution requirements, namely how the ontologies were represented (section 3.1), how the ontologies were integrated (section 3.2), and how SPARQL queries were used to analyze the integrated ontologies (section 3.3).

3.1 Ontology Representation

In order to fulfill requirement **R1**, ontologies representing the Archimate, COBIT 5 PRM and Metrics ontologies had to be specified using the OWL language.
Ontologies are specifications of conceptualizations, meaning that ontologies are suitable to specify conceptual schemas [Studer et al., 1998]. Such schemas depict abstract views over a given domain, conceiving the necessary concepts and relations to describe it. Ontologies enable the representation of vocabularies and enable the classification of instances according to such definitions.

When building an ontology, it is important to be clear why it is being built and what its intended uses are [Uschold and Gruninger, 1996]. As such, in order to represent each of the required ontologies, we had to perform a series of steps. First, a careful analysis of such domains and the extraction of its main concepts and relations, followed by its representation in a meta-model. Then, having the meta-model, a transformation of its concepts and relations to OWL classes and properties was the next step. Finally, the definition of a set of axioms restricting the ontology elements was also performed.

The next two sections describe these steps applied to the representation of COBIT 5 PRM and Metrics, respectively. Note that, as mentioned in section 2.6, work towards the representation of the Archimate ontology as already been performed [Bakhshadeh et al., 2014]. As such we reused the existing Archimate ontology.

3.1.1 COBIT 5 PRM

This research work explores the use of ontologies as means to support the evidence collection for COBIT 5 process capability level 1 assessment. To fulfill this purpose we have identified the need of representing the COBIT 5 PRM as it includes the necessary indicators to assess a process capability level. Actually, when determining the capability level of a given process, the documentation used is the one regarding COBIT 5 PAM, which includes process performance and capability indicators. However, as mentioned before, we have only addressed the assessment of capability level 1, which relies only on the process performance indicators. These can be found on PAM, which for capability level 1 describes them using the COBIT 5 PRM, only augmenting it with information regarding the support of practices and work products to process goals.

Following the steps described previously, based on COBIT 5 documentation [ISACA, 2012a, ISACA, 2012b, ISACA, 2013], and based on previous work towards the specification of COBIT PRM meta-model [Goeken and Alter, 2009, Souza Neto and Ferreira Neto, 2013, Moeller et al., 2013, Textor and Geihs, 2015], we extracted the main concepts and relations which are summarized in table 3.1 and represented on the UML class diagram depicted in figure 3.2.

Having the meta-model specified, its concepts and relations were transformed into OWL classes and properties. We used the standard language OWL 2, profile DL, in order to perform such activities. OWL 2 provides two types of properties, namely *DataTypeProperties* which relate a given object with a literal value (integers, booleans, among others), and *ObjectProperties*, which relate objects with other objects instance of a certain class. Regarding the depicted meta-model diagram, classes contained in it representing COBIT 5 PRM concepts were transformed into OWL 2 *classes*. Relations between such concepts were transformed into *ObjectProperties*, and each diagram class attribute was transformed to a *DataTypeProperty* whose literal type corresponds to the respective one contemplated on the diagram.

Table 3.1: COBIT 5 PRM concepts and definitions.

Process	Generally, a collection of practices influenced by the enterprise's policies and procedures that takes inputs from a number of sources (including other processes), manipulates the inputs and produces outputs.
Practice	High level requirements for effective and practical management and governance of enterprise IT. The tasks and activities needed to accomplish the process purpose and fulfill the process outcomes. Each BP is explicitly associated to a process outcome.
Activity	The main actions to operate the process . They are defined as 'guidance to achieve management practices for successful governance and management of enterprise IT'.
Process Purpose	A description of the overall purpose of the process
Process Goal	A statement describing the desired outcome of a process. An outcome can be an artifact, a significant change of a state or a significant capability improvement of other processes
IT Related Goal	A statement describing a desired outcome of enterprise IT in support of enterprise goals. An outcome can be an artifact, a significant change of a state or a significant capability improvement.
Enterprise Goal	The translation of the enterprise's mission from a statement of intention into performance targets and results.
Stakeholder Need	Stakeholder needs are influenced by a number of drivers, and can be related to a set of generic enterprise goals.
Metric	Metrics can be defined as 'a quantifiable entity that allows the measurement of the achievement of a goal. Metrics should be SMART— specific, measurable, actionable, relevant and timely'.
Work Product	Artifacts considered necessary to support operation of the process. They enable key decisions, provide a record and audit trail of process activities, and enable follow-up in the event of an incident. They are defined at the governance/management practice level.
Stakeholder	Anyone who has a responsibility for, an expectation from or some other interest in the enterprise. Stakeholders and their responsibility levels are documented in RACI charts.

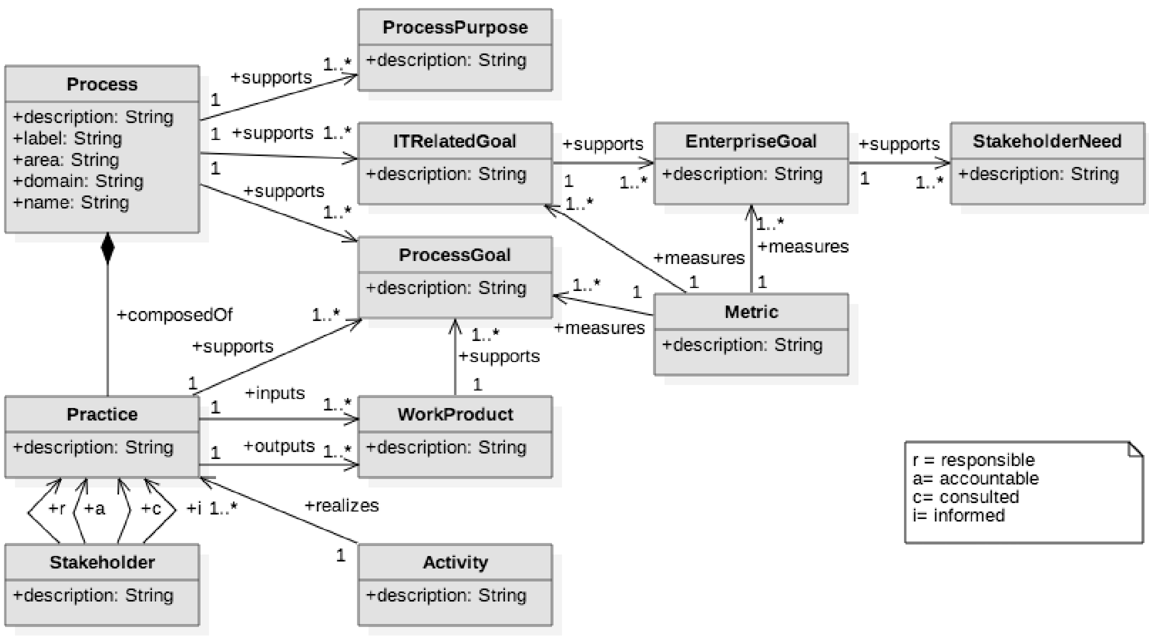

Figure 3.2: COBIT 5 PRM meta-model UML class diagram.

Such transformations were performed using the framework *Protégé*.

To further describe the properties representing the relations between classes, we added a set of axioms. These include relation multiplicity, described by means of OWL cardinality axioms, for example a process is composed of at least one practice. *InverseObjectProperty* and *TransitiveObjectProperty* were also added, enabling a reasoner to infer inverse and transitive properties, respectively. This means, for example, that for the property "supports" it is possible to infer an inverse property named "isSupported". And *TransitiveObjectProperty* provides means to infer that for example if an IT related goal supports and Enterprise goal, and an Enterprise goal supports a Stakeholder need, then an IT related goal supports a Stakeholder need. An excerpt of the resulting COBIT 5 PRM OWL representation can be seen in figure 3.3.

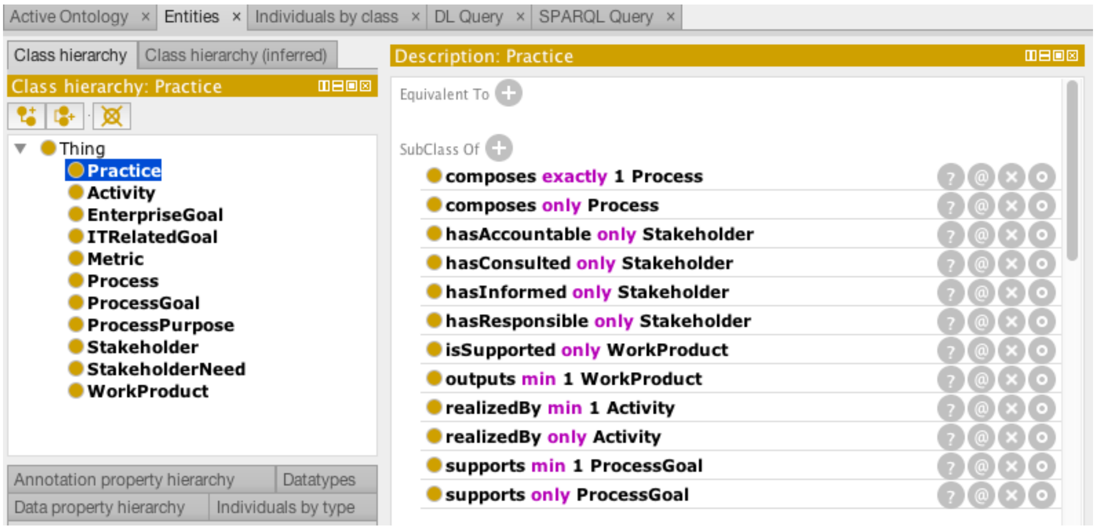

Figure 3.3: COBIT 5 PRM OWL ontology excerpt from *Protégé*.

3.1.2 COBIT 5 Metrics

As for the previous ontology, we now describe the purpose of representing the ontology of metrics. It must be noted that these metrics that we refer to are also described in the COBIT 5 PRM ontology, described through the *Metric* concept. Each metric measures the achievement of a goal, in particular the achievement of a process goal. As mentioned before, in order to verify such achievement we proposed to decompose each metric into a set of properties represented as attributes of an organization's Archimate model's elements. Following this, a metric can be a single property or a combination of multiple properties. As so, the granularity of the concept Metric in the COBIT 5 PRM ontology does not suffice to represent such properties. Therefore, we represented them in the Metrics ontology.

COBIT 5 provides a very large set of example metrics, from which we have only chosen a subset containing **ten**. These metrics belong to processes ***DSS02 - Manage Service Requests and Incidents*** and ***DSS05 - Manage Security Services***, and are described as means to measure these processes goals achievement. We chose these particular processes pertaining to the Deliver Support and Service (DSS) domain, as it refers to the actual delivery of the IT services required to meet strategic and tactical plans, meaning that choosing the architectural concepts to later extend would be a less burden, as such concepts would be more likely to be contemplated in the Archimate language.

As mentioned before, these metrics defined by means of properties would then be integrated with the Archimate meta-model, which we will further explain in the remainder of this document. Following this, the criteria to choose these metrics was to extend as much different Archimate concepts from the three different layers as possible, and to choose at least one by process goal. Therefore the chosen metrics were:

1. Mean elapsed time for handling each type of service request.
2. Level of user satisfaction with service request fulfillment.
3. Number and percent of incidents causing disruption to business-critical processes.
4. Percent of incidents resolved within an agreed-on/acceptable period of time.
5. Number of vulnerabilities discovered.
6. Number of Firewall breaches.
7. Number of incidents involving endpoint devices.
8. Number of Accounts vs Number of Authorized Users.
9. Number of physical security-related incidents.
10. Number of incidents relating to unauthorized access to information.

Having this set, based on the analysis of each of the metrics contained in the chosen subset, we decomposed each metric into a set of possible properties. The majority of the metrics is represented using only a property, but there are some that result from the combination of multiple properties. For example for metric *"3 - Number and percent of incidents causing disruption to business-critical processes"*, the notion of a ***critical*** process as well as the possible types of ***incidents registered by the organization that cause disruption*** were specified as properties to represent such metric. As well as metric *"4 - Percent of incidents resolved within an agreed-on/acceptable period of time"*, which required a property per all possible types of ***incidents registered by the organization resolved within and agreed-on/acceptable period of time***. Also metric *"8 - Number of Accounts vs Number of Users"* required the combination of a property ***number of accounts*** and ***number of authorized users***.

Table 3.2: COBIT 5 DSS02 metrics and properties.

Metric	**Property**
Number and percent of incidents causing disruption to business-critical processes.	*critical* *noOfIncidentsUAccessInfoDisr* *noOfIncidentsPhysicalSecDisr* *noOfIncidentsEDevicesDisr*
Percent of incidents resolved within an agreed-on/acceptable period of time.	*noOfIncidentsResolvedUAccessInfo* *noOfIncidentsResolvedPhysicalSec* *noOfIncidentsResolvedEDevices*
Level of user satisfaction with service request fulfillment.	*levelOfSatisfactionWithSR*
Mean elapsed time for handling each type of service request.	*meanTimeSR*

Each property was represented using the OWL 2 *DataTypeProperty* primitive, and range axioms were also added so that the type of each property was restricted. Most of the represented properties have a range of the literal type *int*, but for example *"critical"* property has a range of the literal type *boolean*, as it is supposed to tag if a business process is critical or not. Tables 3.2 and 3.3 show for process DSS02 and DSS05 each metric and respective properties, respectively .

Table 3.3: COBIT 5 DSS05 metrics and properties.

Metric	**Property**
Number of vulnerabilities discovered.	*noOfVulnerabilities*
Number of firewall breaches.	*noOfBreaches*
Number of accounts (vs. number of authorised users/staff).	*noOfAccounts* *noOfUsers*
Number of incidents involving endpoint devices.	*noOfIncidentsEDevices*
Number of physical security-related incidents.	*noOfIncidentsPhysicalSec*
Number of incidents relating to unauthorised access to information.	*noOfIncidentsUAccessInfo*

3.2 Ontology Integration

In order to fulfill requirement **R2**, the integration of the existing Archimate ontology with COBIT 5 PRM and Metrics ontologies was performed. In section 2.4, we have shown techniques to integrate ontologies, namely the mapping, alignment and merging techniques. We used ontology mapping as means to integrate the ontologies. Recalling the definition of ontology mapping, it consists of a process to build a new ontology by describing the semantic relationship between two (or more) concepts from two (or more) ontologies [Amrouch and Mostefai, 2012, Noy, 2009]. As such, ontology mapping is a set of correspondences between elements of different ontologies, and these correspondences can be equivalence relationships, subclass of superclass relationships, transformation maps, and other more specific correspondences [Noy, 2009].

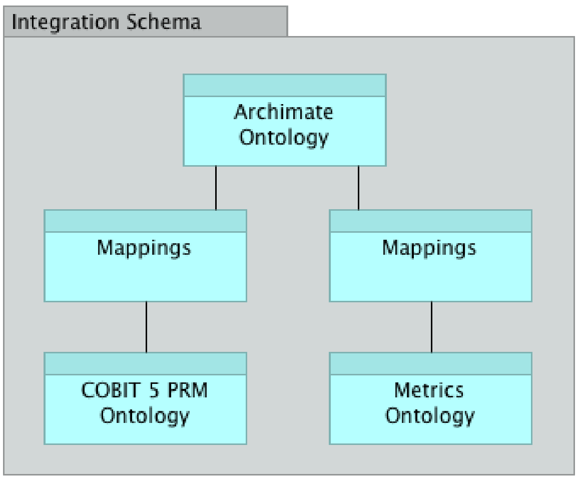

Figure 3.4: The adopted configuration for the ontologies integration.

There may be various configurations for integration schemas. In this particular case, we aim to extend the Archimate's meta-model with necessary concepts to address COBIT 5 assessment concerns. As such, we adopted a similar approach to Antunes et al. [Antunes et al., 2015], in order to address such specific analysis concerns, by defining Archimate as an Upper Ontology (UO) containing general EA concepts and relations, and COBIT 5 PRM and Metrics ontologies as Domain Specific Ontologies (DSO) representing the concepts pertaining to these specific domains. Figure 3.4 depicts the configuration of the adopted integration schema.

Ontology integration is context-dependent, meaning that the mappings defined to integrate ontologies are also designed to address specific concerns. Each DSO was integrated with the UO by specifying a set of ontology mappings, which were then represented using OWL 2 language primitives, in a new ontology. In the next two sections we describe how we have integrated Archimate with COBIT 5 PRM and Metrics ontologies, respectively.

3.2.1 Archimate and COBIT 5 PRM

COBIT 5 PRM, as its name suggests, is a reference model. The COBIT 5 framework's documentation provides the *"what"* should be done in order to achieve certain goals, but not the *"how"* to be done. Each enterprise must implement its own set of processes, using the COBIT 5 processes as a reference and adapting them to its own context and needs. Therefore, we find that enterprise elements contained in Archimate models are realizations of COBIT 5 performance indicators, contained in COBIT 5 PRM. Note also, that the purpose of integrating the Archimate ontology with COBIT 5 PRM ontology is to able to answer COBIT 5 evidence collection questions regarding the existence of Archimate elements realizing COBIT 5 ***Practices***, ***Work Products*** and ***Process Goals***. Hence, such realization relationships must be transformed into a set of mappings.

The Archimate language already delivers a *realization* relationship which fits such purpose, as it models that some end is realized by some means [Standard and Group, 2013]. However, as it is a relationship specific of Archimate, and its use constrained to Archimate concepts only, such a relationship does not contemplate in its range concepts pertaining to COBIT 5 domain. As such, we explored the Archimate's meta-model extension as means to reuse the *realization* relationship covering also COBIT 5 concepts. By meta-model extension we mean the specialization of concepts pertaining to the Archimate's meta-model with concepts pertaining to COBIT 5 PRM meta-model, so that the latter inherits Archimate's restrictions and *realization* mappings are possible.

In order to specify the ontology mappings at the meta-model level we analyzed both Archimate and COBIT 5 PRM meta-models, and based on existing work towards such integration, we extracted semantic relationships between both meta-models concepts represented as classes. As these mappings purpose is to be able to detect which Archimate elements realize which COBIT 5 performance indicators, we constrained the COBIT 5 PRM concepts to map to a subset containing: *Process Goal*, *Practice*, *Work Product*.

As mentioned in section 2.8, work towards the ontological mapping of Archimate and COBIT 5 PRM meta-models was already performed [Almeida et al., 2016, Cadete, 2015]. Almeida et al. in order to map, model and integrate COBIT 5 with COSO governance framework, considered that such frameworks should be based on the same language [Almeida et al., 2016]. As such in order to provide a uniform representation creating a common frame of reference, both frameworks concepts were mapped to Archimate concepts. The ontological mapping of COBIT 5 PRM to Archimate was based on both Archimate's Business layer and Motivation extension.

Table 3.4: COBIT 5 performance indicators mapped to Archimate concepts adapted from Almeida et al. and Cadete.

COBIT 5 concept	[Almeida et al., 2016]	[Cadete, 2015]
Process Goal	Goal	Goal
Practice	Business Process	Requirement
Work Product	Business Object	Requirement

Also, Cadete produced an ontological mapping of COBIT 5 PRM to Archimate, in order to incorporate the COBIT 5 rationale into Archimate as means to assist COBIT 5 process assessments and process improvements initiatives [Cadete, 2015]. The author has also represented such ontological mapping based on both Archimate's Business layer and Motivation extension, but with a major focus on the latter. An excerpt of both authors ontological mappings, containing only the subset of selected concepts, is presented on table 3.4, in which the two right-most columns represent the Archimate concepts mapped to each COBIT 5 concept.

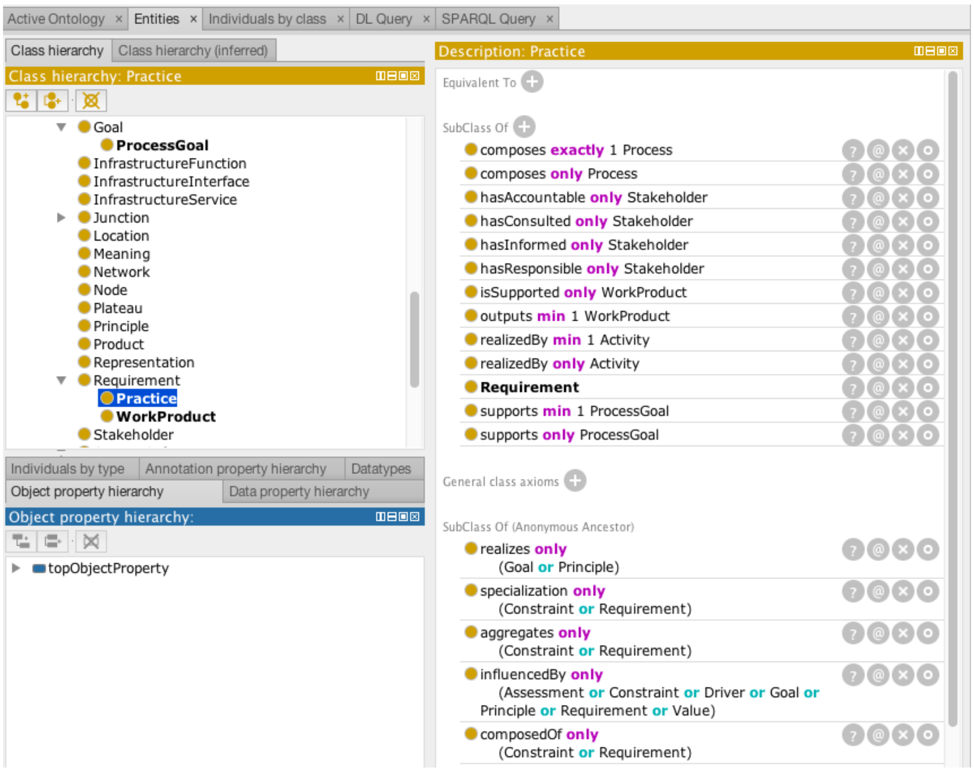

Figure 3.5: Excerpt of the OWL integrated ontology of Archimate and COBIT 5 PRM from Protégé.

We opted to use the ontological mapping of author Cadete [Cadete, 2015], as it better fits our analysis needs. According to its work, performance indicators are mapped as Archimate concepts from the Motivation extension, more precisely as goals and requirements, while according to Almeida et al. [Almeida et al., 2016], the indicators are mapped as core and motivational concepts: business process, business object and goal. Based on the definitions provided in subsection 3.1.1, we find that practices and work products are requirements, rather than business processes and business objects, respectively. Following this, we have transformed the chosen ontology mappings to OWL 2 primitives.

The first approach was to define ontology mappings at the meta-model level as equivalence properties, but we found that concepts from COBIT 5 PRM's meta-model are rather specializations of Archimate's meta-model. Thus, we used the OWL 2 primitive *subClassOf* to represent the specializations. The resulting ontology after the mappings can be seen in figure 3.5. As the figure depicts, a *subClassOf* property was added to the COBIT 5 *Practice* in order to specialize the Archimate concept *Requirement*. Having such a property, the constraints applied to Archimate relationships are also applied to the concept *Practice*, because the latter inherits its *superClass* axioms. The same applies to the remaining COBIT 5 concepts mapped, meaning that it is now possible to use *realization* relationships between Archimate ontology core concepts instances and COBIT 5 performance indicators.

3.2.2 Archimate and Metrics

We now describe how we mapped the properties represented on the Metrics ontology to Archimate concepts. Note that the primary purpose of representing such properties was to be able to verify the conformity of architectural elements, so that COBIT 5 process goals fulfillment could be checked. Hence, in order to extend the Archimate's meta-model with the represented properties, we first analyzed the metrics that such properties compose along with the two processes they belong to, and the concepts pertaining to the Archimate meta-model.

Recalling the processes whose metrics we haven chosen: *DSS02 - Manage Service Requests and Incidents* and *Manage Security Services*; both belong to COBIT 5 DSS domain and they are both concerned with service delivery. Metrics from process *DSS02* are concerned with how incidents impact on user satisfaction, business processes and services. While metrics from process *DSS05* are concerned with network and communications security and incidents at technology infrastructure level. Hence why for *DSS02* process metrics we have used concepts pertaining to the three Archimate layers, while for *DSS05* process metrics we had a major focus on the Archimate's *Technology* layer.

Based on such analysis we formulated the mappings between Archimate's meta-model concepts and Metrics properties, which can be found in tables 3.5 and 3.6, for metrics pertaining to process *DSS02* and *DSS05*, respectively. In order to represent such mappings, we have used OWL 2 domain restriction axioms, so that the Archimate's meta-model extension could be performed. Meaning that for each OWL *DataTypeProperty* representing a given Metric's property, we constrained its domain to the Archimate's meta-model concept according to aforementioned mappings. An excerpt of the resulting mappings can be seen in figure 3.6.

Table 3.5: COBIT 5 DSS02 metrics properties mappings to Archimate concepts.

Property	Archimate Concept	Archimate Concept Definition
critical	*Business Process*	"A behavior element that groups behavior based on an ordering of activities. It is intended to produce a defined set of products or business services."
noOfIncidentsUAccessInfoDisr noOfIncidentsPhysicalSecDisr noOfIncidentsEDevicesDisr	*Node*	"A computational resource upon which artifacts may be stored or deployed for execution."
	Device	"A hardware resource upon which artifacts may be stored or deployed for execution."
noOfIncidentsResolvedUAccessInfo noOfIncidentsResolvedPhysicalSec noOfIncidentsResolvedEDevices	*System Software*	"A software environment for specific types of components and objects that are deployed on it in the form of artifacts"
levelOfSatisfactionWithSR	*Business Actor*	"An organizational entity that is capable of performing behavior."
meanTimeSR	*Application Service*	"A service that exposes automated behavior."
	Infrastructure Service	"An externally visible unit of functionality, provided by one or more nodes, exposed through well-defined interfaces, and meaningful to the environment."

Table 3.6: COBIT 5 DSS05 metrics properties mappings to Archimate concepts.

Property	**Archimate Concept**	**Archimate Concept Definition**
noOfVulnerabilities	*Network*	"A communication medium between two or more devices."
noOfBreaches	*Node*	"A computational resource upon which artifacts may be stored or deployed for execution."
noOfIncidentsEDevices		
	Device	"A hardware resource upon which artifacts may be stored or deployed for execution."
noOfIncidentsPhysicalSec		
	System Software	"A software environment for specific types of components and objects that are deployed on it in the form of artifacts."
noOfIncidentsUAccessInfo		
noOfAccounts *noOfUsers*	*Application Component*	"A modular, deployable, and replaceable part of a software system that encapsulates its behavior and data and exposes these through a set of interfaces."

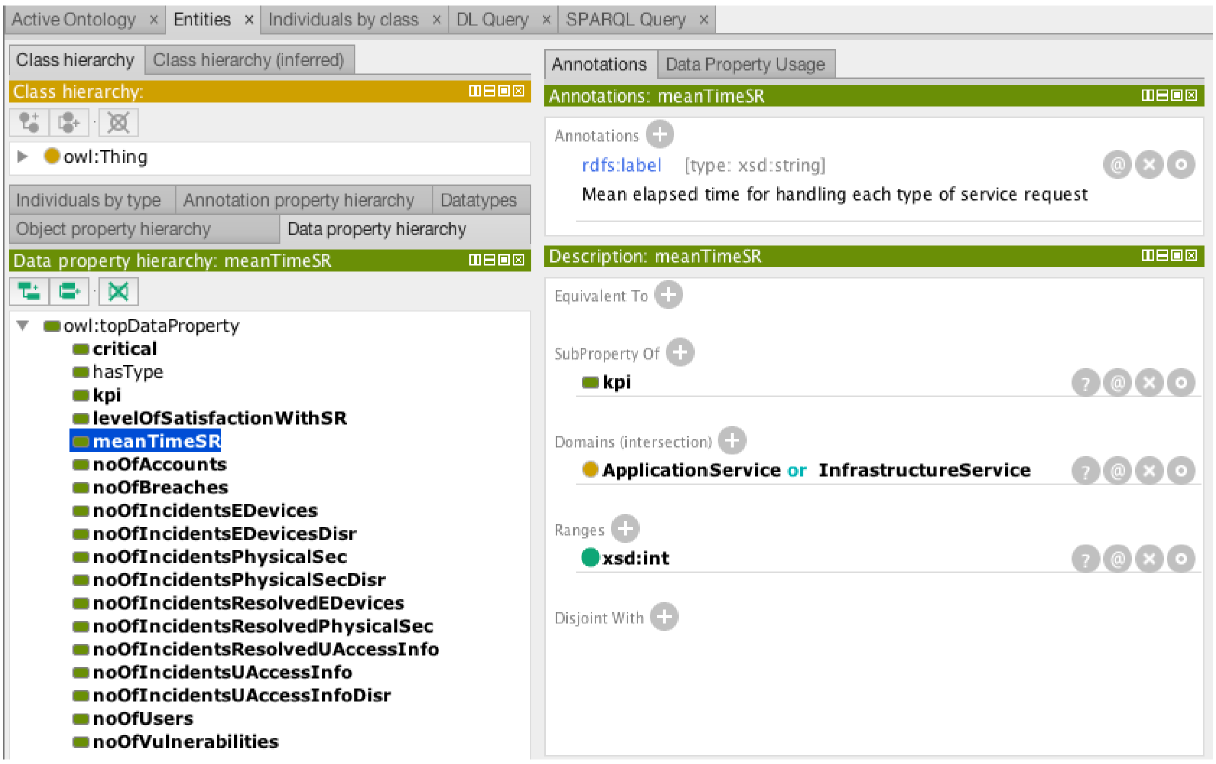

Figure 3.6: Excerpt of the OWL integrated ontology Archimate's concepts and Metrics properties from Protégé.

3.3 Ontology Analysis

In the beginning of this research proposal, we stated the requirement **R3 - The solution must provide means to automatically analyze the integrated ontology based on SPARQL queries, providing answers to questions that support evidence collection for COBIT 5 process capability level 1 assessment**. By now we have already represented the required ontologies, as well as we have provided means to integrate them. Hence, we now lack of a set of the so mentioned evidence collection questions and the respective SPARQL queries to answer them. The questions should reflect the approaches described in section 3, and answers to such questions should be delivered by generating views over the earlier described integrated ontology. Hence, viewpoints to generate these views are required, and were defined by means of SPARQL queries. In the subsequent subsections we present a set of possible stakeholder questions that map to the above mentioned approaches, and the queries that provide answers to such questions.

3.3.1 Stakeholders Questions

Based on the described approaches and on the represented ontologies, we formulated a set of example stakeholders questions, which are listed bellow. Note that these are not the only possible questions that can be asked. We found that these could be relevant, and answers to these questions could be of importance in support of evidence collection for COBIT 5 process capability level 1 assessment. We have also defined some questions we found interesting, regarding the COBIT 5 goals cascade tool, and also questions in order to retrieve rankings of architectural elements. Hence, the defined stakeholder questions were:

1. Given a **Stakeholder Need**, which **Enterprise Goals support** it?
2. Given a set of **Enterprise Goal**, which **Process**/es **support** them, ordered by decreasing number of support relations?
3. Given a **Process**, what **Process Goals** does it **support**?
4. For each **Process Goal**, which **Archimate elements realize** it?
5. Is there any **Archimate element** which **realizes** at least one **Process Goal**, and whose performance is **measured** through **Metrics**?
6. For each **Process Goal**, which **Archimate elements realize** it and whose performance is not **measured** through **Metrics**?
7. For each **Process Goal** which **Metrics** are used to **measure** its fulfillment?
8. Which **Network** elements exceed the acceptable **number of vulnerabilities**, and which **Process Goal** do they **realize**?
9. Which **Node/Device/System Software** elements exceed the acceptable **number of firewall breaches**, and which **Process Goal** do they **realize**?
10. Which **Application Component** elements have a **number of accounts** bellow the **number of authorized users**, and which **Process Goal** do they **realize**?
11. Which **Node/Device/System Software** elements exceed the acceptable **number of Endpoint Device incidents**, and which **Process Goal** do they **realize**?

12. Which **Node**/**Device** elements exceed the acceptable **number of physical-security related incidents**, and which **Process Goal** do they **realize**?

13. Which **Node**/**Device**/**System Software** exceed the acceptable **number of incidents relating to unauthorized access to information**, and which **Process Goal** do they **realize**?

14. Which **Business Service**/**Application Service**/**Infrastructure Service** elements exceed the acceptable **mean time for handling service requests**, and which **Process Goal** do they **realize**?

15. Which **Business Actor** elements do not have an acceptable **level of satisfaction regarding service request fulfillment**,and which **Process Goal** do they **realize**?

16. Which **critical Business Process** elements have a **number of incidents causing disruption** exceeding the acceptable, and which **Process Goal** do they **realize**?

17. Which **Archimate elements** have a **percentage of resolved incidents according to agreed-on service levels** exceeding the acceptable, and which **Process Goal** do they **realize**?

18. Which **Archimate elements** contribute to the **realization** of **Process Goals**, ordered by number of contributions?

19. For a given **Process**, which **Practices compose** it?

20. For a given **Process**, which **Business Process realize Practices** that **compose** it?

21. For a given **Process**, which **Business Object realize Work Products** that are **input/output** for the **Practice** that **compose** it?

22. For a given **Process** ,which **Practices** are not **realized** by a **Business Process**, and which **Process Goals** do they **support**?

23. For a given **Process**, which **Work Products** are not **realized** by a **Business Object**, and which **Process Goals** do they **support**?

Questions ranging from 1 to 3 provide means to help establish the scope of processes to assess. These questions were built taking into account the COBIT 5 goals cascade tool, and answers to them should contain the processes that support a certain stakeholder need according to such tool.

Questions ranging from 4 to 18 focus on the conformity of enterprise elements contained in the organization's EA representation, and answers to such questions provide stakeholders with a list of the non-compliant elements and the set of COBIT 5 process goals not achieved. Each of these questions refers to an acceptable number, such value is defined by the organization performing the assessment, and should be adjusted to its context and needs.

And finally, questions ranging from 19 to 23 provide means to acknowledge which practices and work products were adopted and produced, respectively. Answers to these questions should contain the architectural elements realizing practices and work products. And, also the practices and work products not adopted or missing, if any. Each question has the necessary concepts and relations to answer it in bold. These concepts and relations will be found in the viewpoints defined by SPARQL queries in the subsection bellow.

3.3.2 SPARQL Queries

Recall that part of the research objective was to provide automatic means to answer a set of questions to support the evidence collection for COBIT 5 assessment, hence why also each of the above defined questions was translated into a SPARQL query. The SPARQL language was designed to query data that conforms to the RDF data model, since OWL relies on RDF, the represented and integrated ontologies can be queried using such language. SPARQL provides a set of primitives such as *SELECT* which retrieves a set of instances that conform to a set of triple patterns defined as conditions and *ASK* which retrieves an answer in the form a boolean value (true or false) if it finds instances conforming to a set of triple patterns. Among others, for example, the primitive *FILTER* which enables the definitions of more specific conditions or *ORDER* which orders a set of selected instances according to alphabetical or numerical criteria.

```
SELECT ?businessProcess (SUM(?value) as ?total_number_of_incidents_causing_disruption) ?
    processGoal
WHERE {
        ?pg rdf:type cobit:ProcessGoal.
        ?bp rdf:type uo:BusinessProcess.
        ?bp kpi:critical \"true\"^^xsd:boolean.
        ?bp rdfs:label ?businessProcess.
        ?bp uo:realizes ?pg.
        ?pg cobit:measured ?m.
        ?pg cobit:description ?processGoal.
        ?e rdfs:label ?element.
        ?e uo:dependsUp ?bp.
        ?e ?dataProperty ?value.
        FILTER (?m = cobit:DSS02.PG.1_Metric_1).
        { ?dataProperty rdfs:label \"Number␣of␣incidents␣involving␣endpoint␣devices␣
    causing␣disruption\"} UNION
        { ?dataProperty rdfs:label \"Number␣of␣physical␣security-related␣incidents␣
    causing␣disruption\"} UNION
        { ?dataProperty rdfs:label \"Number␣of␣incidents␣relating␣to␣unauthorised␣access␣
    to␣information␣causing␣disruption\"}.
        }
        GROUP BY ?businessProcess ?processGoal
        HAVING (?total_number_of_incidents_causing_disruption > 40)
        ORDER BY DESC(?total_number_of_incidents_causing_disruption)
```

Listing 3.1: SPARQL query for question 16

For example, question 16 was translated into a SPARQL query, as can be seen in 3.1. Such query selects three variables, the *business processes* that are tagged as critical, according to the property critical defined in the Metrics ontology and whose total number of incidents causing disruption is superior to a given integer value; the *total number of incidents causing disruption* itself; and it also retrieves the *process goal* that such business processes realize. Note that the total number of incidents causing disruption is calculated by retrieving all the elements upon which the critical *business processes* depend on, and which must have a *DataProperty* assertion of some type of incident that cause disruption.

```
SELECT ?practice ?processGoal
WHERE {
        ?p rdf:type cobit:Practice.
        ?pg rdf:type cobit:ProcessGoal.
        ?p cobit:supports ?pg.
        ?pg cobit:description ?processGoal.
        ?p cobit:name ?practice.
        ?process rdf:type cobit:Process.
        ?p cobit:composes ?process.
        FILTER (?process = cobit:DSS02)
        MINUS {
                ?bp rdf:type uo:BusinessProcess.
                ?bp uo:realizes ?p.
                }
        }
```

Listing 3.2: SPARQL query for question 22

Also, the SPARQL query translating question 22 can be seen in 3.2. Such query selects two variables, in this case a not realized *practice* and the respective *process goal* it supports. In order to retrieve such practice, we first obtain all the practices and then, using the *MINUS* primitive we subtract these with the practices that are realized by *business processes*. In order to run the total set of queries on top of the defined integrated ontologies, we built an application written in Java and which uses the Jena framework. This application loads the ontologies using the Jena Ontology API, and runs the defined queries using the ARQ engine. The results of the queries execution are written as a report.

Chapter 4

Demonstration

This chapter maps to the Demonstration step of DSRM, and in which we demonstrate the utility of our Solution artifact applied to a particular scenario. Such demonstration was performed by executing the described set of SPARQL queries over the defined (populated) ontology resulting from the integration of Archimate, COBIT 5 PRM and Metrics ontologies. In order to perform such demonstration, we rely on *three assumptions*: the existence of an Archimate model representing an organization's EA, the existence of mappings between elements of such model and COBIT 5 performance indicators validated with the organization's stakeholders, and finally the existence of concrete values to populate the represented metrics. Note that the mentioned mappings need not be represented as OWL mappings, we just require its documentation on some kind of format.

We chose to use the Archisurance case study [Jonkers et al., 2012] because of its relevance to the Enterprise Architecture community. Although, the Archisurance case study does not contemplate any adoption of Governance frameworks, such as COBIT 5, its authors encourage its extension by adding new views or change scenarios, as long as these are coherent with the initial case study description. As such we have extended the scenario with an Archimate view containing a set of business processes and business objects regarding the subject of incident management. We have also performed the required mappings, as well as defined arbitrary values to populate the metrics.

The following subsections describe the set of steps undertaken to perform the demonstration:

- section 4.1 briefly describes the Archisurance Case Study and presents the added view regarding incident management;
- section 4.2 describes how we have populated the resulting ontologies and which instance-level mappings we have performed;
- section 4.3 presents a subset of results of applying the defined SPARQL queries in order to answer the respective questions.

4.1 Archisurance Case Study

The Archisurance case study is a fictitious example published by The Open Group [Jonkers et al., 2012], and whose purpose is to illustrate the usage of the Archimate modelling language. The case study is meant to describe the Enterprise Architecture of Archisurance, an insurance company that resulted from the merging of three previously independent insurance companies: Home & Away specialized in homeowners' insurance and travel insurance, PRO-FIT specialized in auto insurance, and Legally Yours specialized in legal expense insurance.

The lead investors of the three companies realized that only a larger combined company, Archisurance, could fulfill a specific set of goals including: *control costs*, *maintain customer satisfaction*, *achieve competitive advantage by investing on IT*, and have a *dynamic response to market changes*. These goals will be later used in the subsection 4.3 in order to acknowledge which COBIT 5 processes to assess.

The case study presents the baseline architecture of the company Archisurance, in which we focus, and also a set of change scenarios. It describes the business architecture emphasizing two main business processes *Handle Claim* and *Close Contract*. It also describes the information systems architecture, providing views over the *Application* and *Technology* layers, and how these impact on business. The *Application* layer contains *Application* components such as *Policy Data Management*, *Customer Data Management*, *Claim Data Management*, and *Financial Application*, among others. The *Technology* layer contains Servers represented as *Nodes* and *Devices*, associated to Firewalls and *Networks*, and which in turn realize a number of *Infrastructure Services*. Figure 4.1 depicts a layered view over the Archisurance EA representation retrieved from.

As mentioned before, the Archisurance case study does not contemplate the adoption of governance or management frameworks. Hence, in order to be able to demonstrate the utility of our solution artifact, we now describe the Archimate view we produced containing business processes and business objects to represent a process of Incident management. Recall that the questions regarding the adoption of COBIT 5 practices and production of COBIT 5 work products, defined in section 3.3.1, require the existence of instances of business processes and business objects in the organization's EA representation that supposedly realize instances of COBIT 5 practices and work products.

We chose these particular process as it was our focus along our research proposal. Based on COBIT 5 documentation [ISACA, 2012b], we represented a set of business processes meant to fulfill a set of required activities to implement incident management, as well as a set of business objects produced and used in such activities. Figure 4.2 depicts the resulting Archimate view. Note that, the COBIT 5 process for incident management is *DSS02 - Manage service requests and incidents*. We have only represented business processes and objects regarding the part of incident management, discarding service request management. Hence such a process is expected, according to our representation, never to achieve capability level 1.

4.2 Ontology Instances

Now that we have extended the Archisurance EA representation with an Archimate view containing business processes and objects regarding Incident Management, it is necessary to populate the ontologies represented and integrated in section 3. We also lack the properties that represent Metrics to be populated with concrete numbers, and also the relations between instances of a populated Archimate ontology with a populated COBIT 5 PRM ontology. The steps required to create instances of such ontologies are further detailed bellow.

Archimate OWL ontology - As we opted to use the Archisurance case study, instances of the Archimate OWL ontology are elements described in Archisurance EA representation, and are classified ac-

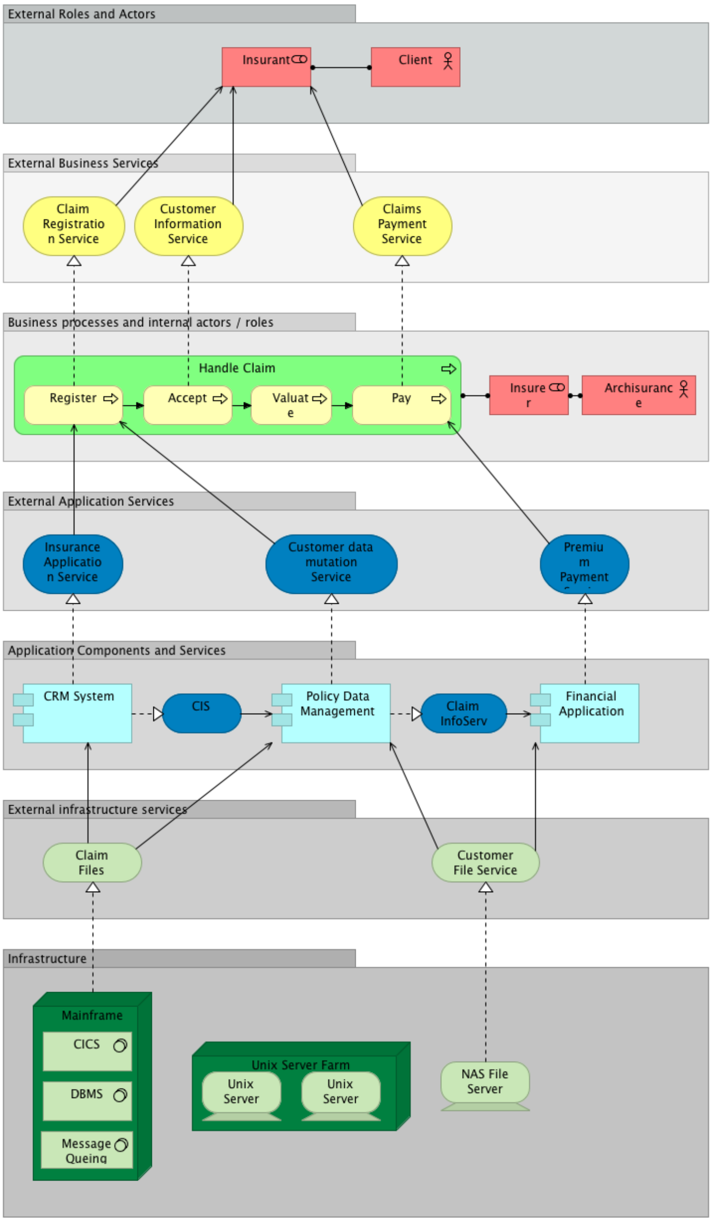

Figure 4.1: Archisurance EA Representation Model's Layered View [Jonkers et al., 2012].

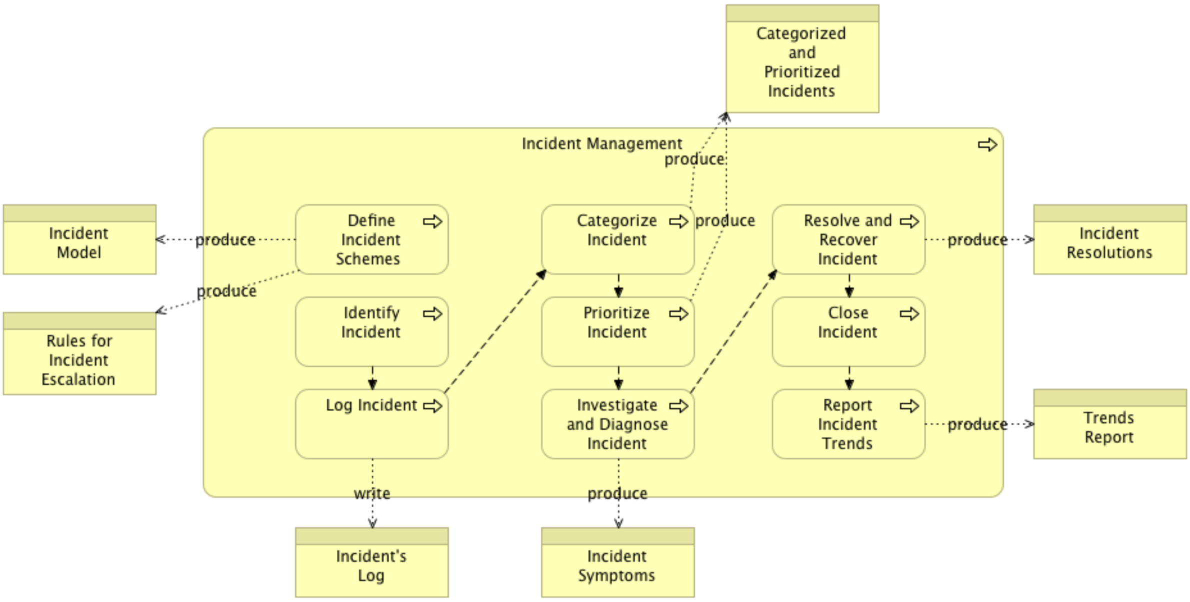

Figure 4.2: Archimate view to address Incident Management concerns.

cording to the concepts pertaining to the ontology. We did not have to perform the step of instance creation for this particular ontology, as it already exists resulting from previous published work towards the analysis of EA models [Bakhshandeh et al., 2013]. The OWL representation of Archisurance contains then the various architectural elements pertaining to the EA representation of the case study as individuals of the Archimate OWL ontology classes.

SPARQL is not able to detect relations described by means of OWL *Property chains*. As such, in order to overcome this limitation we used the Protégé tool to export an inferred model. This feature adds to the ontology a set of inferred and asserted information about the ontology, in the form of inferred axioms, whose type can be chosen according to one's needs. For example property assertions, class assertions, inverse object properties, among others. The inferred axioms are produced by running a reasoner over the selected ontology.

Following this, and using the already existent Archisurance OWL representation, we exported the respective inferred model, and it is now possible to run the defined SPARQL queries that explore the usage of the Archimate OWL ontology *Property chains*: *dependsUp* and *dependsDown*. An excerpt of the inferred axioms for business process *"Pay"* can be seen in figure 4.3, in the property assertions tab. Also regarding figure 4.3, in the left-most tab, instances of the added Incident management business processes can be seen.

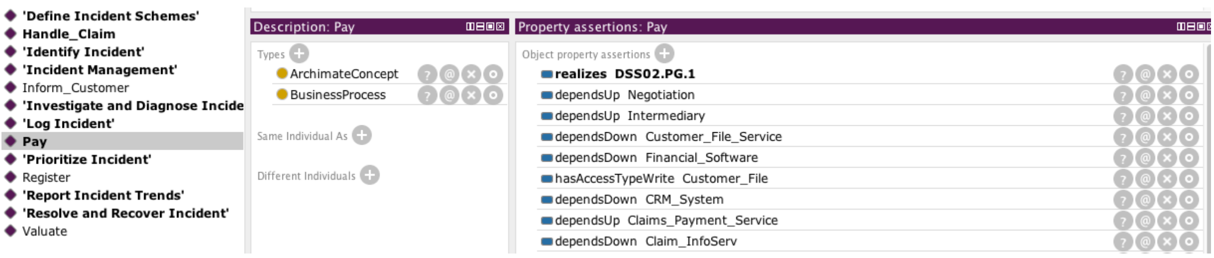

Figure 4.3: Excerpt of Archisurance OWL inferred model in Protégé.

COBIT 5 OWL ontology - In order to populate the COBIT 5 ontology, we built a Java based program to automatically create the required instances according to the classes and relations described in the ontology. Note that ISACA provides the contents of COBIT 5 PRM in CSV format for its members. Hence, our program contains a module that parses the CSV files, and according to the file schemas and the COBIT 5 ontology elements, creates the individuals for the ontology classes and relates them by creating *ObjectProperty* assertions.

The Java program uses the *Jena* framework, more precisely, it uses the Ontology and RDF API, in order to import, load, and populate the COBIT 5 PRM ontology. Information regarding concepts such as *enterprise goals*, *stakeholder needs* and RACI charts was not contemplated in the CSV files provided by ISACA. As such, we manually created CSV files containing that information, which were also parsed by our program in order to create the required instances. The populated ontology contains the whole COBIT 5 PRM 37 seven processes, along with all their related information, resulting in 2444 individuals, and about 47748 logical axioms. Figure 4.4 depicts an excerpt of the populated COBIT 5 PRM ontology.

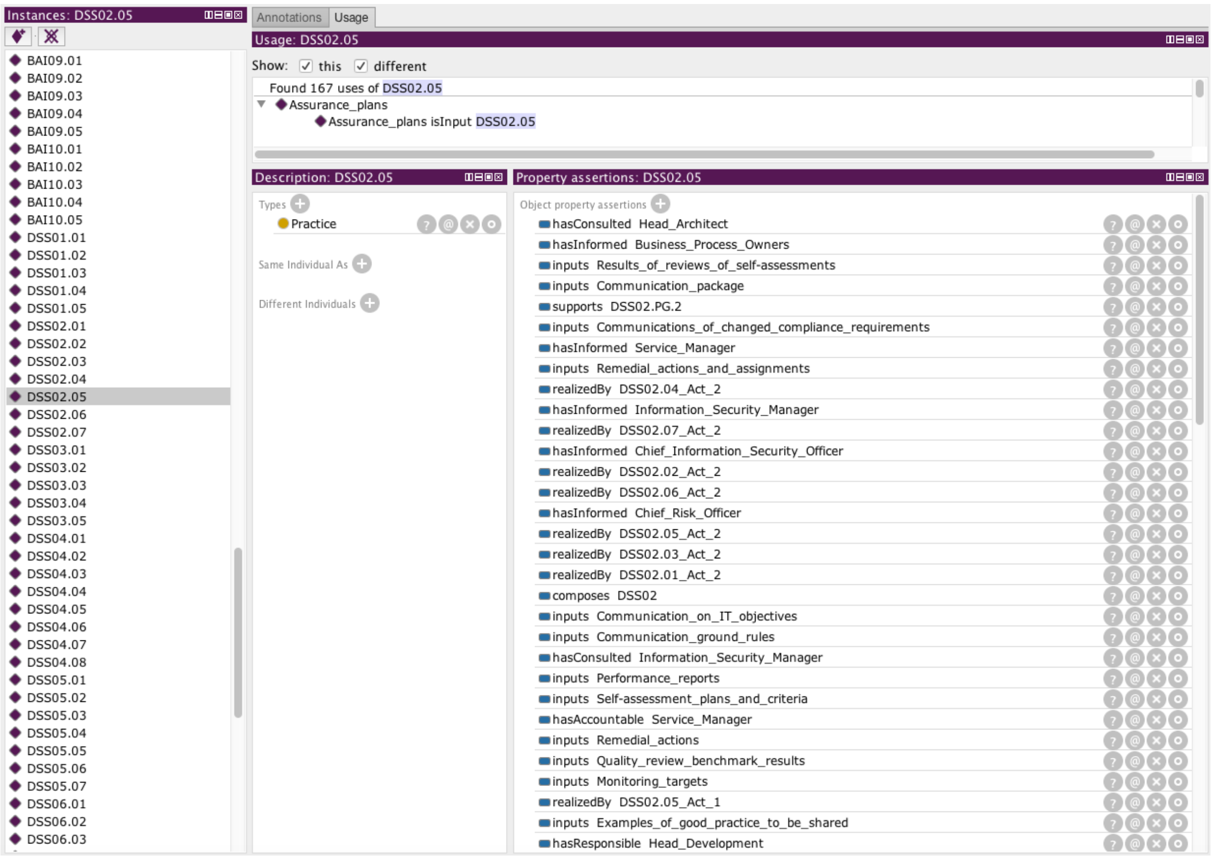

Figure 4.4: Excerpt of COBIT 5 OWL populated ontology in Protégé.

Metrics OWL ontology and mappings - In this particular ontology we did not create instances, as properties that compose the metrics are represented with *DataTypeProperties*. Following this, and as such properties are mapped to Archimate OWL ontology classes, we added *DataProperty* assertions to instances of the Archisurance OWL representation. For each Archisurance OWL ontology individual, classified according to the respective Archimate OWL classes that were mapped to the Metrics ontology properties represented as *DataProperties*, we added an *DataProperty* assertion with arbitrary values.

For instance, in figure 4.5 it can be seen the Archisurance OWL individual "BIBIT" of Archimate OWL class *Network* with a property assertion for the *DataTypeProperty* "noOfVulnerabilities" with the integer value of 23. The same process was performed to a subset of the remaining Archisurance OWL ontology individuals.

Additionally, as mentioned earlier, a set of instance-level mappings was performed, as the analysis we shall further execute demands the existence of such relations. Therefore, for each Archisurance OWL individual, to which we added a *DataProperty* assertion, we also added an *ObjectProperty* assertion in order to relate such instance by means of the Archimate's *realization ObjectProperty* with instances of COBIT5 PRM ontology, more precisely, instances of the class *ProcessGoal*, as depicted in figure 4.5. Moreover, for each Archisurance OWL instance of the Archimate's OWL classes *BusinessProcess* and *BusinessObject* related to incident management, we added a set of *ObjectProperty* assertions in order to relate such instances by means of Archimate's *realization ObjectProperty* with instances of COBIT 5 PRM classes *Practice* and *WorkProduct* related to COBIT 5 PRM process *DSS02*, respectively.

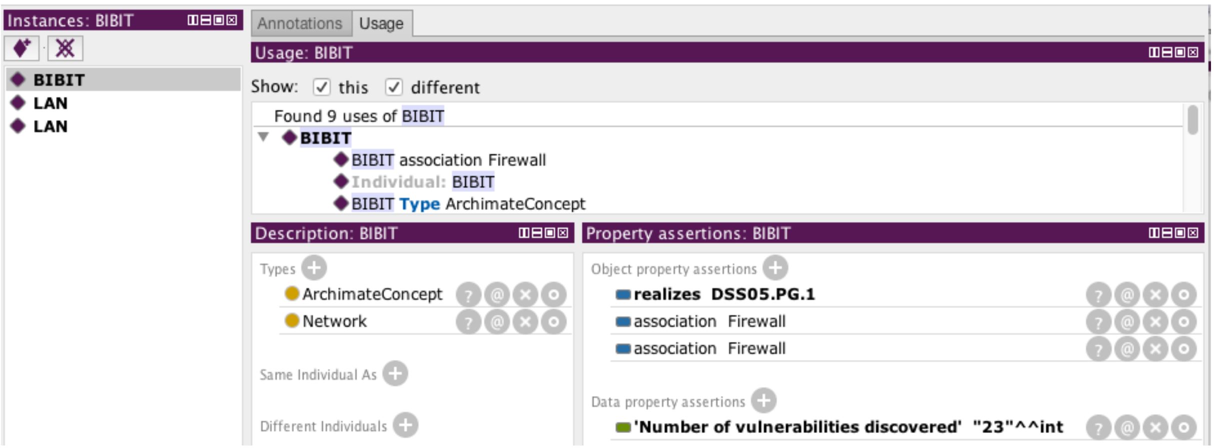

Figure 4.5: Excerpt of Metrics OWL populated ontology in Protégé.

4.3 Ontology Analysis

Having the populated ontologies, we now demonstrate the analysis capabilities of the developed SPARQL queries. Each of the defined SPARQL queries, which translate a set of defined stakeholder questions, was executed. These queries and the respective results can be retrieved from the public *GitHub* repository [1]. Some of the stakeholder questions defined in section 3.3 required explicit values to be specified in order to execute the correspondent query. For example, question 8 *"Which **Network** elements exceed the acceptable **number of vulnerabilities**, and which **Process Goal** do they **realize**?"* required the specification of the acceptable number of vulnerabilities, as well as the remaining questions regarding compliance metrics verification. Also stakeholder question 1, which required the specification of the stakeholder needs to be addressed, and also stakeholder question 2 which relies on the results from question 1.

Following this, each of the stakeholder questions that needed specific values for variables was adapted. Stakeholders needs and enterprise goals were inferred from the Archisurance's case study description, and each "acceptable value" was substituted with an arbitrary value. Hence we now present an excerpt

[1] https://github.com/miggyana/Thesis.git

of the results from the queries executed. Note that the questions presented bellow are an excerpt of the correspondent questions defined in section 3.3, but with specific values assigned. The queries results will be further analyzed in section 5.

Question 1 - Given a **Stakeholder Need** (*SN1* and *SN2*), which **Enterprise Goals support** it?

Question 2 - Given a set of **Enterprise Goal** (*EG1*, *EG2*, *EG6*, *EG7* and *EG8*) , which **Process**/es **support** them, ordered by decreasing number of support relations?

Question 8 - Which **Network** elements exceed the acceptable (20) **number of vulnerabilities**, and which **Process Goal** do they **realize**?

Question 11 - Which **Node**/**Device**/**System Software** elements exceed the acceptable (50) **number of Endpoint Device incidents**, and which **Process Goal** do they **realize**?

Question 16 - Which **critical Business Process** elements have a **number of incidents causing disruption** exceeding the acceptable (40), and which **Process Goal** do they **realize**?

Question 20 - For a given **Process** (*DSS02*), which **Business Process realize Practices** that **compose** it?

Question 21 - For a given **Process** (*DSS02*), which **Business Object realize Work Products** that are **input**/**output** for the **Practice** that **compose** it?

Question 22 - For a given **Process** (*DSS02*), which **Practices** are not **realized** by a **Business Process**, and which **Process Goals** do they **support**?

Question 23 - For a given **Process** (*DSS02*), which **Work Products** are not **realized** by a **Business Object**, and which **Process Goals** do they **support**?

```
-------------------------------------------------------------------------------------------
| enterpriseGoal                   | description                                          |
===========================================================================================
| <http://cobit/processes#E_G_1>   | "Stakeholder value of business investments"          |
| <http://cobit/processes#E_G_13>  | "Managed business change programmes"                 |
| <http://cobit/processes#E_G_16>  | "Skilled and motivated people"                       |
| <http://cobit/processes#E_G_17>  | "Product and business innovation culture"            |
| <http://cobit/processes#E_G_2>   | "Portfolio of competitive products and services"     |
| <http://cobit/processes#E_G_6>   | "Customer-oriented service culture"                  |
| <http://cobit/processes#E_G_7>   | "Business service continuity and availability"       |
| <http://cobit/processes#E_G_8>   | "Agile responses to a changing business environment" |
-------------------------------------------------------------------------------------------
```

Figure 4.6: Results from Query 1 execution.

process	enterpriseGoals
<http://cobit/processes#APO01>	5
<http://cobit/processes#APO02>	5
<http://cobit/processes#APO03>	5
<http://cobit/processes#APO04>	5
<http://cobit/processes#APO05>	5
<http://cobit/processes#APO07>	5
<http://cobit/processes#APO08>	5
<http://cobit/processes#APO09>	5
<http://cobit/processes#APO10>	5
<http://cobit/processes#APO11>	5
<http://cobit/processes#APO12>	5
<http://cobit/processes#BAI01>	5
<http://cobit/processes#BAI02>	5
<http://cobit/processes#BAI03>	5
<http://cobit/processes#BAI04>	5
<http://cobit/processes#BAI05>	5
<http://cobit/processes#BAI06>	5
<http://cobit/processes#BAI07>	5
<http://cobit/processes#DSS01>	5
<http://cobit/processes#DSS02>	5
<http://cobit/processes#DSS03>	5
<http://cobit/processes#DSS04>	5
<http://cobit/processes#DSS06>	5
<http://cobit/processes#EDM01>	5
<http://cobit/processes#EDM02>	5
<http://cobit/processes#EDM05>	5
<http://cobit/processes#MEA01>	5
<http://cobit/processes#APO06>	4
<http://cobit/processes#APO13>	4
<http://cobit/processes#BAI08>	4
<http://cobit/processes#BAI10>	4
<http://cobit/processes#EDM04>	4
<http://cobit/processes#BAI09>	3
<http://cobit/processes#EDM03>	3
<http://cobit/processes#DSS05>	2
<http://cobit/processes#MEA02>	2
<http://cobit/processes#MEA03>	2

Figure 4.7: Results from Query 2 execution.

network	number_of_vulnerabilities	processGoal
"BIBIT"	"23"	"Networks and communications security meet business needs."

Figure 4.8: Results from Query 8 execution.

element	number_of_incidents	processGoal
"BIBIT_Server"	"65"	"Information processed on, stored on and transmitted by endpoint devices is protected."
"Mainframe"	"55"	"Information processed on, stored on and transmitted by endpoint devices is protected."
"NAS_File_Server"	"52"	"Information processed on, stored on and transmitted by endpoint devices is protected."

Figure 4.9: Results from Query 11 execution.

businessProcess	total_number_of_incidents_causing_disruption	processGoal
"Pay"	236	"IT-related services are available for use."
"Create_Contract"	78	"IT-related services are available for use."

Figure 4.10: Results from Query 16 execution.

practice	businessProcess
"Define incident and service request classification schemes."	"Define Incident Schemes"
"Record, classify and prioritise requests and incidents."	"Identify Incident"
"Record, classify and prioritise requests and incidents."	"Log Incident"
"Record, classify and prioritise requests and incidents."	"Categorize Incident"
"Record, classify and prioritise requests and incidents."	"Prioritize Incident"
"Investigate, diagnose and allocate incidents."	"Investigate and Diagnose Incident"
"Resolve and recover from incidents."	"Resolve and Recover Incident"
"Close service requests and incidents."	"Close Incident"
"Track status and produce reports."	"Report Incident Trends"

Figure 4.11: Results from Query 20 execution.

workProduct	businessObject
"Incident status and trends report"	"Trends Report"
"Incident resolutions"	"Incident Resolutions"
"Classified and prioritised incidents and service requests"	"Categorized and Prioritized Incidents"
"Incident symptoms"	"Incident Symptoms"
"Incident and service request log"	"Incident's Log"
"Rules for incident escalation"	"Rules for Incident Escalation"
"Incident and service request classification schemes and models"	"Incident Model"

Figure 4.12: Results from Query 21 execution.

practice	processGoal
"Verify, approve and fulfil service requests."	"Service requests are dealt with according to agreed-on service levels and to the satisfaction of users."

Figure 4.13: Results from Query 22 execution.

workProduct	processGoal
"Criteria for problem registration"	"IT-related services are available for use."
"Approved service requests"	"Service requests are dealt with according to agreed-on service levels and to the satisfaction of users."
"Fulfilled service requests"	"Service requests are dealt with according to agreed-on service levels and to the satisfaction of users."
"Problem log"	"Service requests are dealt with according to agreed-on service levels and to the satisfaction of users."
"Closed service requests and incidents"	"Service requests are dealt with according to agreed-on service levels and to the satisfaction of users."
"User confirmation of satisfactory fulfilment or resolution"	"Service requests are dealt with according to agreed-on service levels and to the satisfaction of users."
"Request fulfilment status and trends report"	"Service requests are dealt with according to agreed-on service levels and to the satisfaction of users."

Figure 4.14: Results from Query 23 execution.

Chapter 5

Evaluation

This chapter maps to the DSRM step of evaluation, and in which we assess the proposed solution artifact. According to Peffers et al. such step involves the comparison of the solution objectives and the actual results obtained from the use of the artifact in the demonstration [Peffers et al., 2007]. Hence, for evaluation purposes we have performed an analysis over the demonstration results. Along with such analysis, we have also used a subset of criteria according to Prat et al. DSR artifact evaluation [Prat et al., 2014], and we have also used the four principles of Österle et al. [Österle et al., 2011].

Recalling the established solution objective:

To provide an ontology-based systematic approach capable of answering questions that support evidence collection for COBIT 5 process capability level 1 assessment, based on the automatic analysis of Archimate models enhanced with COBIT 5 properties.

In order to attain the established objective, we proposed an ontology-based approach, in which we described an integrated ontology of Archimate's meta-model and COBIT 5 PRM and metrics meta-models, relying on a set of ontology mappings, and where we have also described how to analyze the resulting integrated ontology based on a set of SPARQL queries. Such integration was proposed as means to address the analysis specificity required to answer the defined example-questions for evidence collection, and as means to overcome the problem acknowledged in section 1.2.

Note that, our solution artifact is the ontology-based approach, which comprises a set of ontologies, integrated by means of ontology mappings, and a set of SPARQL queries that must be able to answer questions that support the evidence collection for COBIT 5 process assessment based on such integration. Hence, we evaluate how effective this approach is in fulfilling such purpose, regardless the correctness/completeness of the meta-models and mappings used. It is not the goal of this research to make that judgment. For instance if a different meta-model is specified for any of the represented ontologies, or if different mappings are specified, the syntax of the proposed queries must be adapted, but we are still able to answer the example-questions.

5.1 Demonstration Results Analysis

We now analyze the results obtained from the Archisurance's case study demonstration, regarding the effectiveness of answers retrieved by the SPARQL queries in support of evidence collection for COBIT 5 process capability level 1 assessment. Recall that we have defined 23 example questions, but only questions ranging from 8 to 17 and from 20 to 23 shall be commented, as they directly support the described approaches in section 3, which in turn are meant to support the evidence collection approaches described by COBIT 5.

Starting by queries ranging from 8 to 17, the results from their execution can either be empty or contain the set of architectural elements represented in the organization's Archimate model that are not compliant according to each of the used metrics. Each metric measures the achievement of a process goal; hence for a given query result if it is empty we can affirm that according to the metric checked, its respective process goal is achieved. As well as the contrary, if the result is not empty, according to the metric checked, its respective process goal is not achieved.

In this particular case, the demonstration results for such queries (8 to 17) show that for both processes *DSS02* and *DSS05*, according to the used metrics, their process goals are not achieved, since there are no empty results. As such, for these set of queries (8 to 17), evidence of process goal achievement (or not) is retrieved, and an assessor can use such information to make its judgments and rate each process performance attribute.

Results from queries 20 and 22, similarly for queries 21 and 23, are complementary. For instance, the results obtained from the demonstration for query 20 contain the set of business processes that realize *DSS02* process practices, while results from query 22 contain *DSS02* process practices that are not realized at all. As such, if an empty result for query 20 is retrieved, then the result for query 22 is certainly not empty, and vice versa. The same happens for queries 21 and 23, but instead of containing business processes and practices, contain business objects and work products, respectively.

For queries 20 and 21, their results suggest that process goals supported by the realized practices and work products are achieved, as the business processes and business objects contained in such results provide evidence that the intent of the practices is being performed and that work products are produced, respectively. While for queries 22 and 23, their results suggest that process goals supported by the non-realized practices and work products are not achieved, since there is no evidence in the organization's EA representation that business processes were implemented or that business objects were produced that realize such indicators.

For results obtained from the execution of queries 22 and 23, there is a need of additional human analysis. We are able to signal that the intent of practices contained in such results is not performed, and that the work products also contained in such results are not produced. However, we are not able to affirm with certainty such fact, as according to COBIT 5 documentation practices and work products may not fit the organization's context, meaning that they may not have been adopted, instead of neglected. Hence, given such information, the assessor must analyze if the signaled practices and work products are in fact missing, taking into account the organization's context, and then rate the process performance attribute according to such judgment.

Table 5.1: Demonstration results interpretation.

Question	Results
Question 1	Result contains the set of enterprise goals that map to the given stakeholder needs.
Question 2	Result contains the set of processes that support the given set of enterprise goals, ordered by the most supporting process to the least.
Question 3	Result contains the set of process goals that support the given processes.
Question 4	Result contains the Archimate elements that support each of the previous process goals.
Question 5	Result contains a boolean value of true, indicating that exists Archimate elements that realize process goals and whose performance is not measured by metrics.
Question 6	Result contains the Archimate elements that realize process goals and whose performance is not measured by metrics.
Question 7	Result contains the metrics used to measure the fulfillment of the given set of process goals.
Question 8	Result contains a Network element not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 9	Result contains the Node elements not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 10	Result contains the Application Component elements not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 11	Result contains the Node/Device elements not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 12	Result contains the Node/Device elements not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 13	Result contains the Node/Device elements not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 14	Result contains the Application/Infrastructure Service elements not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 15	Result contains the Business Actor elements not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 16	Result contains the Business Process elements not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 17	Result contains the Archimate element not compliant, according to the used metric, and the respective not fulfilled process goal.
Question 18	Result contains a set of Archimate elements that realize Process Goals and the respective number of Process Goals realized.
Question 19	Result contains the Practice elements that compose the given Process.
Question 20	Result contains Practice elements and the Business Process elements that realize them.
Question 21	Result contains Work Product elements and the Business Object elements that realize them.
Question 22	Result contains a supposedly missing Practice element and the Process Goal element it supports. Human analysis is required.
Question 23	Result contains a supposedly missing Work Product elements and the Process Goal elements they support. Human analysis is required.

5.2 DSR Evaluation

Based on the above performed analysis of the demonstration's results, we now evaluate our proposed solution artifact according to Prat et al. DSR evaluation hierarchy criteria and the overall research according to Österle et al. principles for DSR in IS [Prat et al., 2014, Österle et al., 2011].

5.2.1 Artifact evaluation in DSR by Prat et al.

Prat et al. has proposed, based on general systems theory, a hierarchy of evaluation criteria for IS artifacts organized according to the fundamental dimensions of a system (goal, environment, structure, activity, and evolution) [Prat et al., 2014]. For each of these dimensions a set of criteria, further detailed in sub-criteria, were defined, from which we chose a subset of two. To assess our solution artifact, we chose the criteria *Goal - **Efficacy***, and *Environment - Consistency with people - **Utility***. We have selected such criteria, because we found these were the most fitting to our research, and also because efficacy and utility are recognized as relevant criteria [Hevner et al., 2004, Peffers et al., 2007, Prat et al., 2014].

Efficacy refers to the degree to which the solution artifact produces its desired effect, i.e. achieves its goal. While ***Utility*** refers to the quality of the solution artifact in practical use. Following this, and applying these two criteria to our solution artifact, we come up with the following results:

- *Goal - **Efficacy** :* The solution provides an integration schema, based on ontology mappings, capable of addressing the analysis concerns related to evidence collection for process capability level 1 assessment. The solution also provides, based on SPARQL queries, means to automatically answer a set of questions to support such activity.

- *Environment - Consistency with people - **Utility** :* The solution is useful as it provides automatic means to assist the evidence collection process that supports process assessment, providing meaningful information for assessors, retrieved from an Archimate model represented as an ontology and enhanced with COBIT 5 concepts and properties. The solution also provides a more formal approach, relying on models represented as ontologies, to support such activity.

5.2.2 Four principles of Österle et al.

Österle et al. claims that scientific research in general needs to be characterized by a set of principles so that it is distinguished from the way solutions are developed in the practitioners' community (e.g. in user organizations) or by commercial providers [Österle et al., 2011]. According to the author, DSR in IS must comply with four basic principles [Österle et al., 2011]:

- **Abstraction :** *Each artifact must be applicable to a class of problems* - In this particular research work abstraction is justified by the possibility of applying the demonstration to any existing Archimate model and the COBIT 5 PRM, transformed into individuals of the proposed integrated ontology.

- **Originality :** *Each artifact must substantially contribute to the advancement of the body of knowledge* - As for as we know, from our literature review, there is no work previously performed towards the ontological integration and analysis of Archimate and COBIT 5 domains as means to support evidence collection for COBIT 5 assessment.

- **Justification :** *Each artifact must be justified in a comprehensible manner and must allow for its validation* - The proposed solution is grounded in previously published work, and on official COBIT 5 documentation. The solution was inferred from the established solution objective, which in turn was inferred based on the research motivation and problem. Every step of the research is described and justified throughout the present document, and it relies on the literature review.
- **Benefit :** *Each artifact must yield benefit, either immediately or in the future, for the respective stakeholder groups* - The proposed solution artifact enables a more formal approach to perform evidence collection for COBIT 5 process assessment, automatically assisting such activity, providing an assessor with relevant information.

Chapter 6

Conclusion

In this research work the solution artifact proposed to solve the earlier acknowledged problem was *"an approach based on the application of a set of ontology-based techniques to represent and integrate Archimate's meta-model with COBIT 5 concepts and properties, and also to automatically analyze the resulting integrated ontology"*. The solution artifact comprises an integration schema for Archimate's meta-model ontology and COBIT 5 PRM and Metrics ontologies, based on a set of ontology mappings, along with a set of SPARQL queries to automatically analyze the resulting integrated ontology.

The integration of the ontologies was meant to overcome the lack of COBIT 5 domain-specific concepts and properties on Archimate's meta-model, necessary to perform an effective model analysis capable of answering questions regarding the evidence collection for process assessment. The set of SPARQL queries provided viewpoints based on such concepts, capable of automatically generating views that address the specific assessment concerns. As means to support such an approach, we specified and represented COBIT 5 PRM and a subset of COBIT 5 example-metrics meta-models as OWL ontologies.

Following this, a Java program was written, using the Jena framework, which receives as input a set of ontologies already populated and mapped, and produces a set of answers based on the execution of queries over the input ontologies. The utility of the solution artifact was demonstrated through the Archisurance case study, and evaluated according to DSRM artifact evaluation and Österle et al. principles.

6.1 Contributions

We now present the contributions attained from this research work, and which we list bellow:

1. An OWL representation for COBIT 5 PRM ontology.
2. An OWL representation for a subset of COBIT 5 example-metrics ontology.
3. An OWL representation for mappings between Archimate's meta-model and COBIT 5 PRM meta-model ontologies.
4. A set of mappings between Archimate and Metrics meta-models ontologies.
5. An OWL representation for mappings between Archimate and Metrics meta-models ontologies.
6. An OWL representation for the populated COBIT 5 PRM meta-model ontology.

7. A set of possible questions to support evidence collection.
8. A set of SPARQL queries to translate the above mentioned questions.
9. A Java application to load ontologies and run queries.
10. A Java application to load, and populate the COBIT 5 PRM ontology.

Note that the main contribution of these research work was the proposed ontology-based approach to answer questions regarding the evidence collection for COBIT 5 process assessment. The above listed contributions appear as a support for such approach.

6.2 Limitations

We have identified some limitations in what concerns the application of the proposed solution, and that we now expose. The execution of the proposed queries over the populated integrated ontology relies on the assumption that the EA representation instantiated in such ontology is a faithful representation, i.e., it correctly depicts the architectural elements comprising the organization's EA. If not, the result obtained from the queries may not fulfill the analysis needs, hence not being able to answer the questions for evidence collection. Thus, incorrect/incomplete Archimate models generate incorrect/incomplete queries results.

The same happens with the instance-level required mappings, as mentioned in section 4, we assume that an organization has these mappings somehow documented and validated with its stakeholders. Again, if not, the results obtained from the queries execution may well be incorrect/incomplete, hence not being able to provide reliable answers to questions regarding the evidence collection. Thus, incorrect/incomplete instance-level mappings generate incorrect/incomplete queries results.

Also, manual transformation of instance-level mappings to OWL primitives may turn out to be very inefficient and time consuming. Note that, the first two limitations are not inferred from our proposed solution, i.e, it is up to the organization to have a faithful representation and validated mappings. Moreover, if the organization adopts the EA practice, and intends to use EA models to support the activity of evidence collection, in theory its architects should be aware of such concerns and provide an EA representation capable of addressing such needs.

6.3 Future Work

As a future work it would be interesting to explore automatic tools for mapping discovery, in order to overcome the issue of mappings manual transformation. For example, automatic tools based on the comparison of names or descriptions. Noy presents possible approaches to reduce the bottleneck introduced by such activity, based on machine-learning or graph-comparison among others [Noy, 2009].

Also, would be interesting to explore the representation of different metrics, for instance the representation of metrics belonging to other COBIT 5 PRM domains instead of *Deliver Service and Support*. Along with experimenting other mappings between ontologies, and other configurations for integration schema.

The population of the Metrics ontology could be automatically performed, using software monitoring tools. For example, for networks management, a software tool to measure the number of vulnerabilities, or the number of firewall breaches could be used in order to populate the represented properties in the Metrics ontology. The same for incident management, as there are incident management tools that collect the required information to populate the represented properties concerning such activity.

Finally, it would be of great value to demonstrate the solution artifact utility and efficacy in a real organization, as in this research we have used a fictitious case study, hence having a controlled environment. The validation of the solution artifact with experts from the COBIT 5 and EA communities through interviews would be also an added value to our research.

Bibliography

[Almeida et al., 2016] Almeida, R., Linares Pinto, P., and Mira da Silva, M. (2016). Using archimate to integrate cobit 5 and coso metamodels. In *European, Mediterranean & Middle Eastern Conference on Information Systems 2016 (EMCIS2016)*, Poland.

[Amrouch and Mostefai, 2012] Amrouch, S. and Mostefai, S. (2012). Survey on the literature of ontology mapping, alignment and merging. In *Information Technology and e-Services (ICITeS), 2012 International Conference on*, pages 1–5.

[Antunes et al., 2014a] Antunes, G., Bakhshandeh, M., Mayer, R., Borbinha, J. L., and Caetano, A. (2014a). Using ontologies for enterprise architecture integration and analysis. *CSIMQ*, 1:1–23.

[Antunes et al., 2015] Antunes, G., Barateiro, J., Caetano, A., and Borbinha, J. (2015). Analysis of Federated Enterprise Architecture Models. *ECIS 2015 Completed Research Papers*, 10:1–14.

[Antunes et al., 2016] Antunes, G., Borbinha, J., and Caetano, A. (2016). An application of semantic techniques to the analysis of enterprise architecture models. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 4536–4545.

[Antunes et al., 2014b] Antunes, G., Caetano, A., and Borbinha, J. (2014b). Enterprise architecture model analysis using description logics. In *Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), 2014 IEEE 18th International*, pages 237–244.

[Bakhshadeh et al., 2014] Bakhshadeh, M., Morais, A., Caetano, A., and Borbinha, J. (2014). *Ontology Transformation of Enterprise Architecture Models*, pages 55–62. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Bakhshandeh et al., 2013] Bakhshandeh, M., Antunes, G., Mayer, R., Borbinha, J., and Caetano, A. (2013). A modular ontology for the enterprise architecture domain. In *2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops*, pages 5–12.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

[Cadete, 2015] Cadete, G. (2015). Using Enterprise Architecture for COBIT 5 Process Assessment and Process Improvement. Master's thesis, Instituto Superior Técnico.

[Caetano et al., 2015] Caetano, A., Antunes, G., Bakhshandeh, M., Borbinha, J., and da Silva, M. M. (2015). Analysis of federated business models: An application to the business model canvas, archimate, and e3value. *2015 IEEE 17th Conference on Business Informatics (CBI)*, 01(undefined):1–8.

[Caetano et al., 2016] Caetano, A., Antunes, G., Pombinho, J., Bakhshandeh, M., Granjo, J., Borbinha, J., and da Silva, M. M. (2016). Representation and analysis of enterprise models with semantic techniques: an application to archimate, e3value and business model canvas. *Knowledge and Information Systems*, pages 1–32.

[Corcho et al., 2006] Corcho, O., Fernández-López, M., and Gómez-Pérez, A. (2006). Ontological engineering: Principles, methods, tools and languages. In *Ontologies for Software Engineering and Software Technology*, pages 1–48. Springer Berlin Heidelberg.

[De Haes and Van Grembergen, 2004] De Haes, S. and Van Grembergen, W. (2004). It governance and its mechanisms. *Information Systems Control Journal*, 1:27–33.

[De Haes et al., 2013] De Haes, S., Van Grembergen, W., and Debreceny, R. S. (2013). Cobit 5 and enterprise governance of information technology: Building blocks and research opportunities. *Journal of Information Systems*, 27(1):307–324.

[DuCharme, 2011] DuCharme, B. (2011). *Learning SPARQL*. O'Reilly Media, Inc.

[Goeken and Alter, 2009] Goeken, M. and Alter, S. (2009). Towards conceptual metamodeling of IT governance frameworks approach - use - benefits. pages 1–10.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition - Special issue: Current issues in knowledge modeling*, 5(2):199–220.

[Guarino et al., 2009] Guarino, N., Oberle, D., and Staab, S. (2009). *What Is an Ontology?*, pages 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Q.*, 28(1):75–105.

[Hitzler et al., 2009] Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., and Rudolph, S., editors (2009). *OWL 2 Web Ontology Language: Primer*. W3C Recommendation. Available at `http://www.w3.org/TR/owl2-primer/`.

[ISACA, 2012a] ISACA (2012a). *COBIT 5: A Business Framework for The Governance and Management of Enterprise IT*. ISACA.

[ISACA, 2012b] ISACA (2012b). *COBIT 5: Enabling Processes*. ISACA.

[ISACA, 2012c] ISACA (2012c). *COBIT 5: Implementation*. ISACA.

[ISACA, 2013] ISACA (2013). *COBIT 5 Process Assessment Model (PAM): Using COBIT 5*. ISACA.

[Johnson et al., 2004] Johnson, P., Ekstedt, M., Silva, E., and Plazaola, L. (2004). Using enterprise architecture for cio decision-making: On the importance of theory. In *Proceedings of the Second Annual Conference on Systems Engineering Research, 2004*.

[Jonkers et al., 2012] Jonkers, H., Band, I., and Quartel, D. (2012). Archisurance case study.

[Klein, 2001] Klein, M. (2001). Xml, rdf, and relatives. *IEEE Intelligent Systems*, 16(2):26–28.

[Lankhorst, 2009] Lankhorst, M. (2009). *Enterprise Architecture at Work: Modelling, Communication and Analysis (The Enterprise Engineering Series)*. Springer, 2nd ed. edition.

[Lankhorst and van Drunen, 2007] Lankhorst, M. and van Drunen, H. (2007). Enterprise Architecture Development and Modelling. Combining TOGAF and ArchiMate. pages 1–16.

[Lenzerini et al., 2004] Lenzerini, M., Milano, D., and Poggi, A. (2004). Ontology representation & reasoning. *Universit di Roma La Sapienza, Roma, Italy*.

[Miller, 1998] Miller, E. (1998). An introduction to the resource description framework. Technical report.

[Moeller et al., 2013] Moeller, B., Erek, K., Loeser, F., and Zarnekow, R. (2013). How sustainable is cobit 5? insights from theoretical analysis and empirical survey data.

[Niemann, 2006] Niemann, K. D. (2006). *From Enterprise Architecture to IT Governance*. Vieweg.

[Noy, 2009] Noy, N. F. (2009). *Ontology Mapping*, pages 573–590. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Österle et al., 2011] Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., and Sinz, E. J. (2011). Memorandum on design-oriented information systems research. *European Journal of Information Systems*, 20(1):7–10.

[Peffers et al., 2007] Peffers, K., Tuunanen, T., Rothenberger, M., and Chatterjee, S. (2007). A design science research methodology for information systems research. *J. Manage. Inf. Syst.*, 24(3):45–77.

[Pinto et al., 1999] Pinto, H. S., Gómez-Pérez, A., and Martins, J. P. (1999). Some issues on ontology integration. In *16th International Joint Conference on Artificial Intelligence (IJCAI'99) Workshop: KRR5: Ontologies and Problem-Solving Methods: Lesson Learned and Future Trends*, volume 18. IJCAI and the Scandinavian AI Societies. CEUR Workshop Proceedings. Ontology Engineering Group - OEG.

[Prat et al., 2014] Prat, N., Comyn-Wattiau, I., and Akoka, J. (2014). Artifact evaluation in information systems design-science research – a holistic view. *18th Pacific Asia Conference on Information Systems, Chengdu, China.*, page 16.

[Shadbolt et al., 2006] Shadbolt, N., Berners-Lee, T., and Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101.

[Souza Neto and Ferreira Neto, 2013] Souza Neto, J. A. and Ferreira Neto, A. N. (2013). Metamodel of the it governance framework COBIT. *JISTEM - Journal of Information Systems and Technology Management*, 10:521 – 540.

[Standard and Group, 2013] Standard, O. G. and Group, T. O. (2013). *Open Group Standard The Open Group*.

[Studer et al., 1998] Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data Knowl. Eng.*, 25(1-2):161–197.

[Textor and Geihs, 2015] Textor, A. and Geihs, K. (2015). Calculation of cobit metrics using a formal ontology. In *10th IFIP/IEEE International Workshop on Business-driven IT Management*, pages 1384–1390, Ottawa, Canada.

[Uschold and Gruninger, 1996] Uschold, M. and Gruninger, M. (1996). Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, 11:93–136.