UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
DEPARTAMENTO DE INFORMÁTICA
MESTRADO EM INFORMÁTICA

CARLOS LINS BORGES AZEVEDO

AN ONTOLOGY-BASED SEMANTICS FOR THE MOTIVATION EXTENSION TO ARCHIMATE

VITÓRIA – ES, Brazil

JANUARY 2012

CARLOS LINS BORGES AZEVEDO

AN ONTOLOGY-BASED SEMANTICS FOR THE MOTIVATION EXTENSION TO ARCHIMATE

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

VITÓRIA – ES, Brazil

JANUARY 2012

CARLOS LINS BORGES AZEVEDO

AN ONTOLOGY-BASED SEMANTICS FOR THE MOTIVATION EXTENSION TO ARCHIMATE

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

Aprovada em 27 de janeiro de 2012.

COMISSÃO EXAMINADORA

____

Prof. Dr. João Paulo Andrade Almeida
Universidade Federal do Espírito Santo (UFES) (Orientador)

____

Prof[a]. Dr[a]. Roberta Lima Gomes,
Universidade Federal do Espírito Santo (UFES)

____

Prof[a]. Dr[a]. Fernanda Araujo Baião Amorim
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

____

Prof. Dr. Marten van Sinderen
University of Twente (UT)

VITÓRIA, JANUARY 2012

DEDICATION

To my mother, who with unconditional love, made possible my studies.

ACKNOWLEDGEMENTS

First of all, I want to thank João Paulo Almeida, my supervisor, one of the most brilliant people I have had the pleasure to work with. Thanks especially for your support and help on the work developed in this thesis. I also like to thanks Marten van Sinderen and Dick Quartel, whom have played an important part on the work developed in this thesis and have helped, supervised and supported me to perform the work presented here. Other very especial person with a high impact on the work presented here is Giancarlo Guizzardi, who introduced me to ontologies and conceptual modeling, when I was still doing my Computer Science course.

I couldn't forget to thank my NEMO friends. A non-exhaustive list of them includes: Paulo Sérgio, Julião, Tiago Prince, Pedro Paulo, Veruska, Calhau, Lucas, Ana Christina, among various others. You all have been a great support for conversations and off course, UFO, ontologies and conceptual modeling doubts reasoning.

And finally, for the ones who are not reading this thesis for research purposes (ok, ok, they will be reading just this part), I would like to thank them in Portuguese. Um agradecimento especial à minha família, minha mãe, minha irmã e meu pai. O núcleo familiar é o que possibilita que nos tornemos o que somos hoje. Quero também agradecer à minha avó, que sempre vibrou por mim.

Por último, agradeço a todos os que estiveram comigo nesse processo, aos amigos que eu fiz e que ficarão para sempre, aos amigos que aqui já estavam e que por aqui se mantiveram, enfim, a todos que me apoiaram neste processo.

ABSTRACT

Keywords: Enterprise Architecture, Enterprise Architecture Modeling, Ontology-based Semantics, Goal Modeling, TOGAF, ArchiMate, UFO.

The "motivation domain" of an Enterprise Architecture addresses objectives in a broad scope ranging from high-level statements expressing the goals of an enterprise to declarations of requirements on business processes, services and systems. An important development regarding the incorporation of the motivation domain in a comprehensive Enterprise Architecture modeling language is the upcoming Motivational Extension to ArchiMate (based on the ARMOR language). The extension proposes the inclusion of concepts such as concerns, assessments, goals, principles and requirements to ArchiMate.

Similarly to other goal modeling approaches, the initial development of the Motivation Extension has been conducted without a formal definition of its concepts. We believe that careful definition of the semantics of these concepts is required, especially when considering that the motivation domain addresses subjective aspects of the enterprise. To address that, this thesis focuses on an ontology-based semantics for the Motivation Extension. The concepts are interpreted by using the Unified Foundational Ontology (UFO) as a semantic foundation, and, as a result, propose well-founded recommendations for improvements of the extension.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Artificial Intelligence……………………………………………………………………...…AI

Business Process Modeling Notation ....BPMN

Enterprise Architecture ....EA

Information Technology ……………………………………………………………………… IT

Motivation Extension to ArchiMate...ME

Object Management Group ....OMG

The Open Group Architecture Framework …..………………………………………TOGAF

Unified Foundational Ontology..UFO

CONTENT

CHAPTER 1. INTRODUCTION

1.1 BACKGROUND

Managing an organization is a complex task, since it relies on different domains of knowledge (going from business processes, information technology (IT) and infra-structure to the organization’s motivation, goals and concerns) (DODAF, 2007). The increasing competitiveness drives organizations to promote changes, adapting and improving in order to continue in business. Predicting how a given enterprise environment should respond to changes is currently a great concern and is a challenging question, especially since a change in an organization strategy will probably impact the organization as whole (its business processes, IT, goals, etc.).

To analyze the integration of these organizational domains and how the priorization of one can impact another, a structured description of the enterprise is required. Enterprise architecture enables the organization to control functionality and complexity. An enterprise architecture consists of “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, IACOB, *et al.*, 2005).

Enterprise architecture is the key to lowering the risks and managing the various organizational knowledge domains. That is because the enterprise architecture allows the assessment of investments and the assessment of the impacts of these investments (DODAF, 2007).

Enterprise architecture structures the enterprise in terms of various related architectural domains or viewpoints (IEEE COMPUTER SOCIETY, 2000), (DIJKMAN, 2006) each of them focusing on specific aspects of the enterprise. According to Zachman (1997), this is because “when describing an object, it is convenient to isolate a single characteristic at a time. Attempting to deal with all the characteristics at one time would result in such a complex depiction it would be incomprehensible, useless. It is a process of ‘abstraction’, of extracting out of the total, a more simplistic sub-set on which to focus”. Each of the organizational domains focuses on one major area of the enterprise. The enterprise viewpoints are defined since the Zachman framework (SOWA and ZACHMAN, 1992) and according to it, the organizational viewpoints are (Figure 1 illustrates Zachman viewpoints):[1]

i. The Data viewpoint;
ii. The Function viewpoint;
iii. The Network viewpoint;
iv. The People viewpoint;
v. The Time viewpoint and;
vi. The Motivation viewpoint;

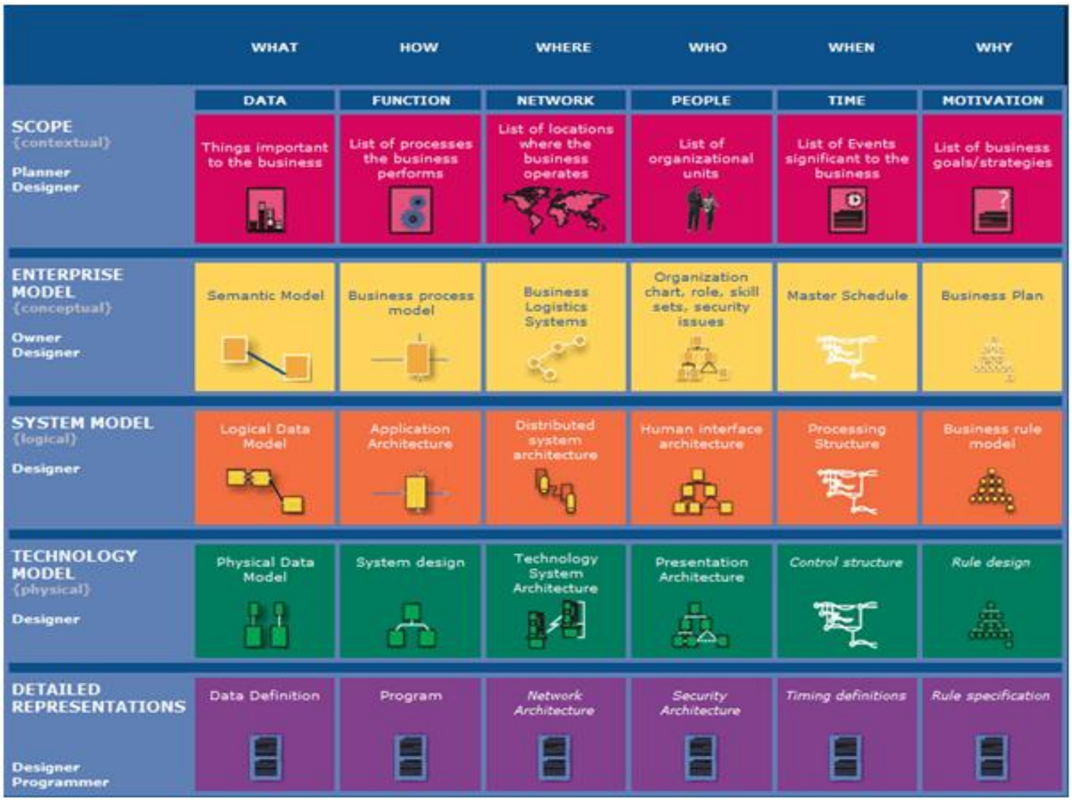


Figure 1-Enterprise Architecture Viewpoints (SOWA and ZACHMAN, 1992)

Each of these domains (or viewpoints) has quality criteria and requirements. Some of these quality criteria and requirements can potentially conflict and they all influence in the organization's performance.

[1] Zachman did not define the Why (Motivational) column in his original, first framework. However, the Why column was deemed relevant and was introduced in 1992 in his reviewed, evolved framework. Since then the Why column is part of the framework.

Among the many domains of an enterprise, the motivation domain is of particularly importance because its statements impact on all the organizational domains. The motivation domain is concerned with "why" an Enterprise Architecture is defined the way it is. This domain addresses enterprise objectives in a broad scope ranging from high-level statements expressing the goals of an enterprise to declarations of requirements on business processes, services and systems (CARDOSO, ALMEIDA and GUIZZARDI, 2010). In such, a change in the motivation domain necessarily impacts the whole enterprise in order to accomplish the changed objectives or requirements.

1.2 MOTIVATION

Although the importance of the motivation domain for Enterprise Architecture has been recognized since at least two decades ago with the inclusion of the "motivation"/"why" column in the Zachman framework (SOWA and ZACHMAN, 1992), Zachman did not define basic concepts for this column, justifying that "there is a scarcity of good examples in the […] motivation column" and stating that "the why column would be comprised of the descriptive representations that depict the motivation of the enterprise, and the basic columnar model would likely be ends-means-ends, where ends are objectives (or goals) and means are strategies (or methods)" (SOWA and ZACHMAN, 1992). So far, the modeling of motivation has been an active area of research in software and requirements engineering (BRESCIANI, PERINI, *et al.*, 2004), (VAN LAMSWEERDE and LETIER, 2004), with few comprehensive enterprise modeling approaches addressing the why column (CARDOSO, SANTOS JR., *et al.*, 2010).

The use of motivation domain artifacts to address the objectives an organization wants to pursue allows the organization to systematically list and control its objectives. Goal modeling explicitly captures the goals of an enterprise, documenting an enterprise strategy (ANDERSSON, BERGHOLTZ, *et al.*, 2008). "The definition of goals is related with objectives […] within the enterprise, goal statements range from high-level concerns in an enterprise (expressing the vision and mission of the organization) to declarations of the values that must be achieved by business process execution on behalf of stakeholders" (CARDOSO, 2010).

These models state which objectives conflict, which needs to be sequenced or composed to enable the organization to achieve an overarching objective and, furthermore, the requirements and means to accomplish those objectives. This systematic control can be based on organizational needs, priorities, indicators or other preferences. This control can base the organizational movements and correct or change the course of the enterprise in order to accomplish and meet its strategy.

An important development regarding the incorporation of the motivation domain in a comprehensive Enterprise Architecture modeling language is the upcoming Motivational Extension to ArchiMate (based on the ARMOR language) (QUARTEL, ENGELSMAN, *et al.*, 2009). To address the motivational aspect of the enterprise, a goal-oriented requirements modeling language named ARMOR was developed (QUARTEL, ENGELSMAN, *et al.*, 2009). The design of the language was driven by extensive analysis of existing works and the language is considered to have a "more sophisticated support" for goal related concepts when contrasted with a number of enterprise modeling techniques (CARDOSO, ALMEIDA and GUIZZARDI, 2010). The ARMOR language has evolved into the proposed Motivation Extension for the ArchiMate framework (QUARTEL, ENGELSMAN and JONKERS, 2011).

The extension proposes the inclusion of concepts such as *concerns*, *assessments*, *goals*, *principles* and *requirements* to ArchiMate, among others. The aim of the extension is to represent the strategies that shape an Enterprise Architecture. It defines relations between the various concepts to capture how changes in enterprise's goals can have significant consequences throughout the enterprise in an attempt to accomplish the changing objectives.

Similarly to other goal modeling approaches, the initial development of the Motivation Extension has been conducted without a formal definition of its concepts. The absence of such careful definition could lead to several modeling and communication problems. For example, different modelers may ascribe different meanings to the same modeling elements.

This is particularly challenging given that an Enterprise Architecture model is a joint effort involving several stakeholders and requiring the integration of several architectural domains into a coherent whole. The lack of common interpretation for the various modeling elements would lead to integration problems in the organization and information misuse, regardless of whether the models adhere syntactically to the same metamodel.

In order for the models (or systems developed based on those models) to function properly when used together, it must be guaranteed that they ascribe compatible meanings to real-world entities of their shared subject domain. That is only possible if all the people interested on the subject, namely stakeholders, pact on the same meaning for the real-world entities represented by the constructs used on the language, i.e., if they all understand the same information when reading the same models. In particular, it is needed to reinforce that they have "compatible sets of admissible situations, whose union (in the ideal case) equals the admissible state of affairs delimited by the conceptualization of their shared subject domain" (GUIZZARDI, 2005). That must apply to the models constructs and also to the real-world entities that are represented by the model constructs. The ability of systems to interoperate (i.e., operate together), while having compatible real-world semantics is known as semantic interoperability (VERMEER, 1997).

As pointed out by Guizzardi (2005), "a modeling language should be sufficiently expressive to suitably characterize the conceptualization of its subject domain". Further, "the semantics of the produced specifications should be clear, i.e., it should be easy for a specification designer to recognize what language constructs mean in terms of domain concepts" and also the specification produced using the language should "facilitate the user in understanding and reasoning about the represented state of affairs".

We argue that careful definition of the semantics of the concepts of a language to represent the motivational domain is required, especially when considering that the motivation domain addresses subjective aspects of the enterprise.

1.3 RESEARCH OBJECTIVES

This thesis focuses on explicitly defining the semantics for the Motivation Extension proposed to ArchiMate. To accomplish it, the concepts of the Motivation Extension are analyzed and interpreted in terms of a foundational ontology, i.e., a formal and ontologically sound system of domain-independent categories. This thesis proposes a semantic for the Motivation Extension concepts.

1.4 APPROACH

With the purpose of defining the semantics for the Motivation Extension proposed to ArchiMate an approach using ontologies and foundational ontology has been used.

Ontologies are used in the thesis as a proposal to solve the semantic issues disclosed in section 1.2. The term ontology comes from a Latin word which means the *study of the existence*. Since the 80 decade there has been a growing interest in the use of foundational ontologies for evaluating and reengineering conceptual modeling languages and methodologies, as in (WAND and WEBER, 1989) and (WAND and WEBER, 1990). Ontology aims to develop theories about, for example, persistence and change, identity, causality, classification and instantiation, among others. In particular, this thesis uses a foundational ontology as a basis to semantically analyze the ArchiMate proposed extension on the motivation domain.

A foundational ontology defines a system of domain-independent categories and their ties which can be used to articulate the conceptualizations of reality. The use of foundational ontologies aims to ensure ontological correctness of the language and the models described on the language. Ontological analysis is performed by considering a mapping between modeling constructs and the concepts in an ontology. "On the one hand, each modeling element can be interpreted using the ontological theory as a semantic domain. On the other hand, concepts of the domain of discourse (captured in the ontological theory) should be represented by modeling elements of the language being considered. According to (WEBER, 1997), there should be a one-to-one correspondence between the concepts in the ontology and modeling elements" (SANTOS JR, ALMEIDA and GUIZZARDI, 2012, submitted). When the correspondence cannot be obtained, the language problems of construct excess - when a notational construct does not correspond to any ontological concept, construct overload - when a single notation construct can represent multiple ontological concepts, construct redundancy - when multiple modeling elements can be used to represent a single ontological concept, and construct deficit - when there is no construct in the modeling language that corresponds to a particular ontological concept, can be identified[2]. Figure 2 illustrates this topic.

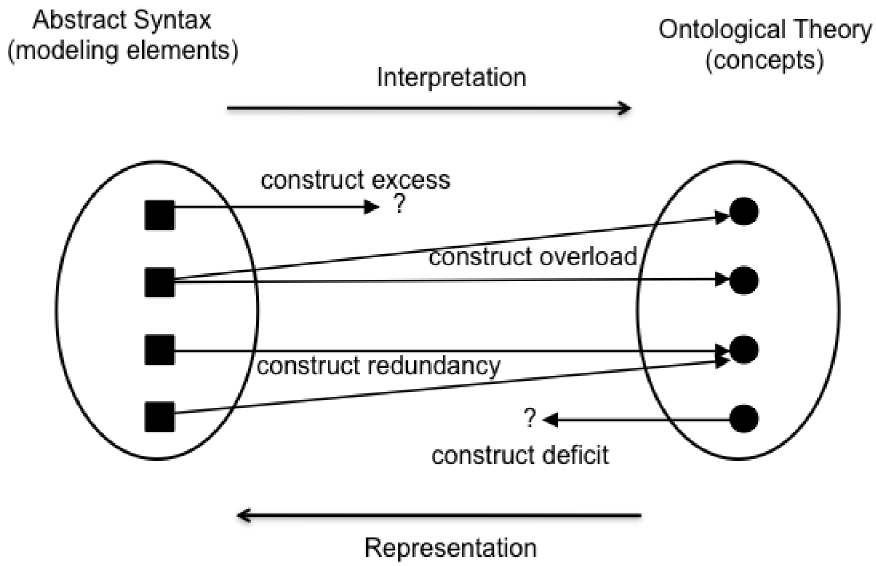


Figure 2- Issues uncovered by ontological analysis – adapted from (MOODY, 2009)

In particular, to uncover these issues and to properly define the semantics of the language, it is used the Unified Foundational Ontology (UFO) as the foundational ontology in this thesis. UFO has been chosen because it unifies several foundational ontologies and has been successfully employed to evaluate, re-design and integrate the models of conceptual modeling languages as well as to provide real-world semantics for their modeling constructs. Further, UFO addresses social and intentional phenomena, which are modeled by the motivation domain. This foundational ontology has been previously used successfully to semantically analyze the Architecture of Integrated Information Systems (ARIS) framework for enterprise modeling (SANTOS JR., ALMEIDA, *et al.*, 2010), to semantically integrate the ARIS framework to the goal modeling language Tropos (CARDOSO, SANTOS JR., *et al.*, 2010), to investigate the semantics of role-related concepts in ArchiMate and a number of Enterprise Modeling approaches (ALMEIDA, GUIZZARDI and SANTOS JR, 2009), and to

[2] The concepts of construct excess, construct overload, construct redundancy and construct deficit are discussed in more detail in section 4.1.

provide the analysis and re-design of agent-oriented goal modeling languages (GUIZZARDI and GUIZZARDI, 2011).

1.5 NON-SCOPE

Due to the fact that the Motivation Extension is still in a proposal status, it is outside the scope of this thesis to analyze the semantics of the Motivation Extension as perceived by its users, an issue which we consider as future work. Such a "pragmatics analysis" would consist in constrasting the semantics given to the language by its users in two cases: (i) when they are working with the language with access to the semantics defined here and (ii) when they only have access to the textual definitions provided by the specification. This study could serve to address the clarity of the semantics defined here. The study would also verify the perception of language users regarding the ontological deficiencies reported on this thesis.

1.6 THESIS STRUCTURE

This thesis is structured in 6 chapters:

i. Chapter 2 (Enterprise Architecture) contextualizes the reader on Enterprise Architectures, especially at TOGAF and at ArchiMate, focusing on its importance for organizations. The intended TOGAF and ArchiMate future extensions and extensions mechanisms are also given and an alignment of TOGAF and ArchiMate is discussed.

ii. Chapter 3 (Ontological Foundation) describes the relevant information on the ontological foundations used to semantically analyze the Motivation Extension to ArchiMate. This chapter introduces the Unified Foundational Ontology and explains the concepts used subsequently on the ontological analysis.

iii. Chapter 4 (The Motivation Extension to ArchiMate) describes the proposed Motivation Extension to ArchiMate. In this chapter the Motivation Extension, its context and concepts are explained. The language metamodel is presented along with the language's concrete syntax. Finally an example using the language is shown and discussed.

iv. Chapter 5 (The Ontology-Based Semantics for the Motivation Extension to ArchiMate) presents the semantic analysis of the Motivation Extension to ArchiMate and its conclusions. In this chapter each of the concepts and relations of the Motivation Extension are analyzed using the Unified Foundational Ontology as its basis. A discussion on each of the concept's analysis is given. Ontological problems found on the concepts (and consequently on the language) are shown and possible solutions or paths to the solution on each of the problems is presented. Then, the example given in chapter 4 is explained in terms of the proposed UFO-based semantics. Finally, the chapter discussed related work.

v. Chapter 6 (Conclusions and Future Work) concludes this thesis. The main contributions are presented and topics for further investigation are identified.

CHAPTER 2. ENTERPRISE ARCHITECTURE

2.1 INTRODUCTION

Enterprise Architecture is defined by Lankhorst et al. (2005) 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".

Enterprise Architecture is a relevant topic under discussion both in academia and in industry. Enterprise Architecture is a powerful and necessary tool to manage the many organizational knowledge domains, where each of these domains is influenced by quality factors and affects the organization's overall performance. These quality factors may potentially conflict and the Enterprise Architecture is needed to allow the balancing and/or prioritization of these factors. In this sense, enterprise architectures are used as instruments to operationalize organizational policies and strategies.

The purpose of enterprise architecture is "to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy" (THE OPEN GROUP, 2011). An enterprise architecture addresses the need of effective management and exploitation of information through Information Technology (IT), since this is an indispensable means to achieve competitive advantage. Enterprise architecture provides a strategic context for the evolution of the IT system in response to the constantly changing needs of the business environment and enables the organization to achieve the right balance between IT efficiency and business innovation (THE OPEN GROUP, 2011), (LANKHORST, IACOB, *et al.*, 2005).

Enterprise architecture helps providing a common basis for the organizational daily operations and is useful to determine its needs and priorities, which can be of changes as well as to determine how the company may benefit from technological operations. Enterprise Architecture captures "the essentials of the business, IT and its evolution. The idea is that the essentials are much more stable than the specific solutions that are found for the problems currently at hand. Architecture is therefore helpful in guarding the essentials of the business, while still allowing for maximal flexibility and adaptability" (JONKERS, LANKHORST, *et al.*, 2006).

Jonkers et al. (2006) stated that the most important characteristic of enterprise architecture is that "it provides a holistic view on the enterprise". Jonkers et al. (2006) also stated that "within restricted domains of expertise that are present in an enterprise, some sort of architectural practice often exists, with varying degrees of maturity. However, due to the heterogeneity of the methods and techniques used to document the architectures, it is very difficult to determine how the different domains are interrelated". A good enterprise architecture provides the insight needed to relate the various domains of the enterprise, pervading from corporate strategy to daily operation. However, in an increasingly global and networked world, no enterprise can focus solely on its own operations. It is indispensable to manage the interconnections within its customers, suppliers, and other partners. Thus, a good enterprise architecture should also facilitate the managing of organizational interconnections.

Commonly, preparation for business transformation needs or for radical infrastructure changes initiates an enterprise architecture review or development. Often areas of change required are identified in order for new business goals to be met.

The role of the architect is to address the concerns introduced above by[3]:

i. Identifying and refining the requirements that key people in the organization have.
ii. Developing views of the architecture that show how the concerns and requirements are going to be addressed.
iii. Showing the trade-offs that are going to be made in reconciling the potentially conflicting concerns of the different key people.

The idea is that without the enterprise architecture, it is highly unlikely that all the concerns and requirements will be considered and met when implementing the organization strategy to achieve the needed transformation.

---

[3] adapted from (THE OPEN GROUP, 2011)

The development of an enterprise architecture is a complex task. To develop a good enterprise architecture a set of methods are a requirement. Enterprise architecture frameworks describe methods for designing the organizational architecture and its information systems. Architecture frameworks help to improve the understanding of the subject by providing systematic approaches to architecture development. These frameworks simplify the architecture's development and ensure coverage of the architectural dimensions. Various frameworks to model enterprise architecture have been developed, such as the Zachman framework (SOWA and ZACHMAN, 1992), the ARchitecture for integrated Information Systems (ARIS) (SCHEER, 2000), the International Organization for Standard Reference Model of Open Distributed Processing (ISO RM-ODP) Enterprise Language (ISO - INTERNATIONAL ORGANIZATION FOR STANDARD, 2008), the Department of Defense Architecture Framework (DODAF) (THE UNITED STATES DEPARTMENT OF DEFENSE) and the The Open Group Architecture Framework (TOGAF) (THE OPEN GROUP, 2011).

One of the most important frameworks is the TOGAF framework, which has been developed through collaborative efforts of 300 member companies and is currently one of the most widely accepted frameworks for Enterprise Architecture (THE OPEN GROUP, 2011). Tang et al. (2004) made a comparative analysis of architecture frameworks and the results have shown that TOGAF was the more comprehensive one of the researched approaches. Table 1 summarizes Tang et al. (2004) comparison, where 'Y' represents 'yes', 'P' represents partially and 'N' represents 'no'.

Table 1- Comparison of Architecture Frameworks - adapted from (TANG, HAN and CHEN, 2004)

	RM-ODP	TOGAF	DoDAF
Goals			
Architecture Definition and Understanding	Y	Y	Y
Architecture Process	N	Y	Y
Architecture Evolution Support	P	Y	Y
Architecture Analysis	Y	Y	Y
Architecture Models	Y	Y	Y
Design Tradeoffs	P	P	Y
Design Rationale	Y	Y	P
Standardization	Y	Y	Y
Architecture Knowledge Base	Y	Y	Y
Architecture Verifiability	P	Y	N
Inputs			
Business Drivers	P	Y	Y
Technology Inputs	P	Y	Y
Business Requirements	Y	Y	Y
Information System Environment	Y	Y	Y
Current Architecture	Y	Y	Y
Non Functional Requirements	Y	Y	P
Outcomes			
Business Model	Y	Y	Y
System Model	Y	Y	Y
Information Model	Y	Y	Y
Computation Model	Y	Y	Y
Software Configuration Model	P	Y	N
Software Processing Model	Y	Y	Y
Implementation Model	Y	Y	Y
Platforms	Y	Y	Y
Non-functional Requirements Design	Y	Y	P
Transitional Design	N	Y	Y
Design Rationale	P	P	P

Even though enterprise architecture frameworks, and particularly TOGAF, define methods for developing an enterprise architecture, a language for the development of the enterprise architecture must also be used. The ArchiMate enterprise architecture modeling language has been developed to “provide a uniform representation for architecture descriptions” (THE OPEN GROUP, 2009). The ArchiMate language offers “integrated architectural approach that describes and visualizes the different architecture domains and their underlying relations and dependencies” (THE OPEN GROUP, 2009). ArchiMate has become a widely adopted language for architecture modeling in the Netherlands and is currently developed by The

Open Group, the same organization that develops TOGAF (THE OPEN GROUP, 2009) and, in such, has various points of integration and can be used concomitantly with TOGAF.

2.2 TOGAF

The Open Group Architecture Framework (TOGAF) is one of the most widely accepted methods for developing enterprise architecture. It was originated as the Technical Architecture for Information Management (TAFIM) in the US Department of Defense. The framework was adopted by the Open Group in the mid-1990s. The first TOGAF specification was introduced in 1995, and currently TOGAF is in its version 9.1 (THE OPEN GROUP, 2011), which was released in December 2011. TOGAF has been developed through the collaborative efforts of over 300 Architecture Forum member companies from some of the world's leading companies and organizations. TOGAF complements, and can be used in conjunction with, other enterprise architecture frameworks that are more focused on specific deliverables for particular vertical sectors such as Government, Telecommunications, Manufacturing, Defense, and Finance.

TOGAF defines an architecture framework as "a foundational structure, or set of structures, which can be used for developing a broad range of different architectures". The TOGAF definition for 'enterprise' is as of "any collection of organizations that has a common set of goals". The term 'enterprise' in the context of 'enterprise architecture' can be used to denote "both an entire enterprise —encompassing all of its information and technology services, processes, and infrastructure —and a specific domain within the enterprise".

The TOGAF architecture definition extends the ISO/IEC 42010: 2007 that defined architecture as "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution". For TOGAF, architecture has two meanings depending upon the context:

i. "A formal description of a system, or a detailed plan of the system at a component level to guide its implementation".
ii. "The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time".

TOGAF is concerned with the definition and the participation of various parts into a whole, i.e., for TOGAF, if the goal is to integrate an extended enterprise, the supply chain is also a part of the enterprise, permitting the closest possible synergy across the extended enterprise.

The TOGAF standard (THE OPEN GROUP, 2011) states that TOGAF covers four related types of architecture. These four types of architectures can be called subsets of an overall enterprise architecture and are:

i. Business Architecture: The "business strategy, governance, organization, and key business processes".
ii. Data Architecture: The "structure of an organization's logical and physical data assets and data management resources."
iii. Application Architecture: A "blueprint for the individual applications to be deployed, their interactions, and their relationships to the core business processes of the organization".
iv. Technology Architecture: The "logical software and hardware capabilities that are required to support the deployment of business, data, and application services. This includes IT infrastructure, middleware, networks, communications, processing, and standards".

The core of TOGAF is the Architecture Development Method (ADM). It describes a method for developing and managing the lifecycle of an enterprise architecture. It integrates elements of TOGAF as well as other available architectural assets, to meet the business and IT needs of an organization.

The TOGAF Architecture Development Method (ADM) provides "a process lifecycle to create and manage architectures within an enterprise". At each phase within the ADM, a discussion of inputs, outputs, and steps describes a number of architectural work products or artifacts, such as a process and application. The TOGAF content metamodel defines a structure for these. Due to the importance of the concepts in the TOGAF content metamodel for the development and use of an enterprise architecture, the next section presents an overview of the TOGAF Content Metamodel. (For a complete discussion on TOGAF and its metamodel the reader should refer to the TOGAF specification (THE OPEN GROUP, 2011).)

2.2.1 Overview of the TOGAF Content Framework

At the highest level, the TOGAF content framework is divided up in 5 parts, or views. Those parts or views are described in line with what TOGAF names ADM phases.

The Architecture Principles, Vision, and Requirements artifacts are intended to "capture the surrounding context of formal architecture models, including general architecture principles, strategic context that forms input for architecture modeling, and requirements generated from the architecture".

The Business Architecture artifacts "capture architectural models of business operation, looking specifically at factors that motivate the enterprise, how the enterprise is organizationally structured, and also what functional capabilities the enterprise has".

The Information Systems Architecture artifacts "capture architecture models of Information Technology systems, looking at applications and data in line with the TOGAF ADM phases".

The Technology Architecture artifacts "capture procured technology assets that are used to implement and realize information system solutions".

The Architecture Realization artifacts "capture roadmaps showing transition between architecture states and binding statements that are used to steer and govern an implementation of the architecture".

Figure 3 shows the TOGAF content framework.

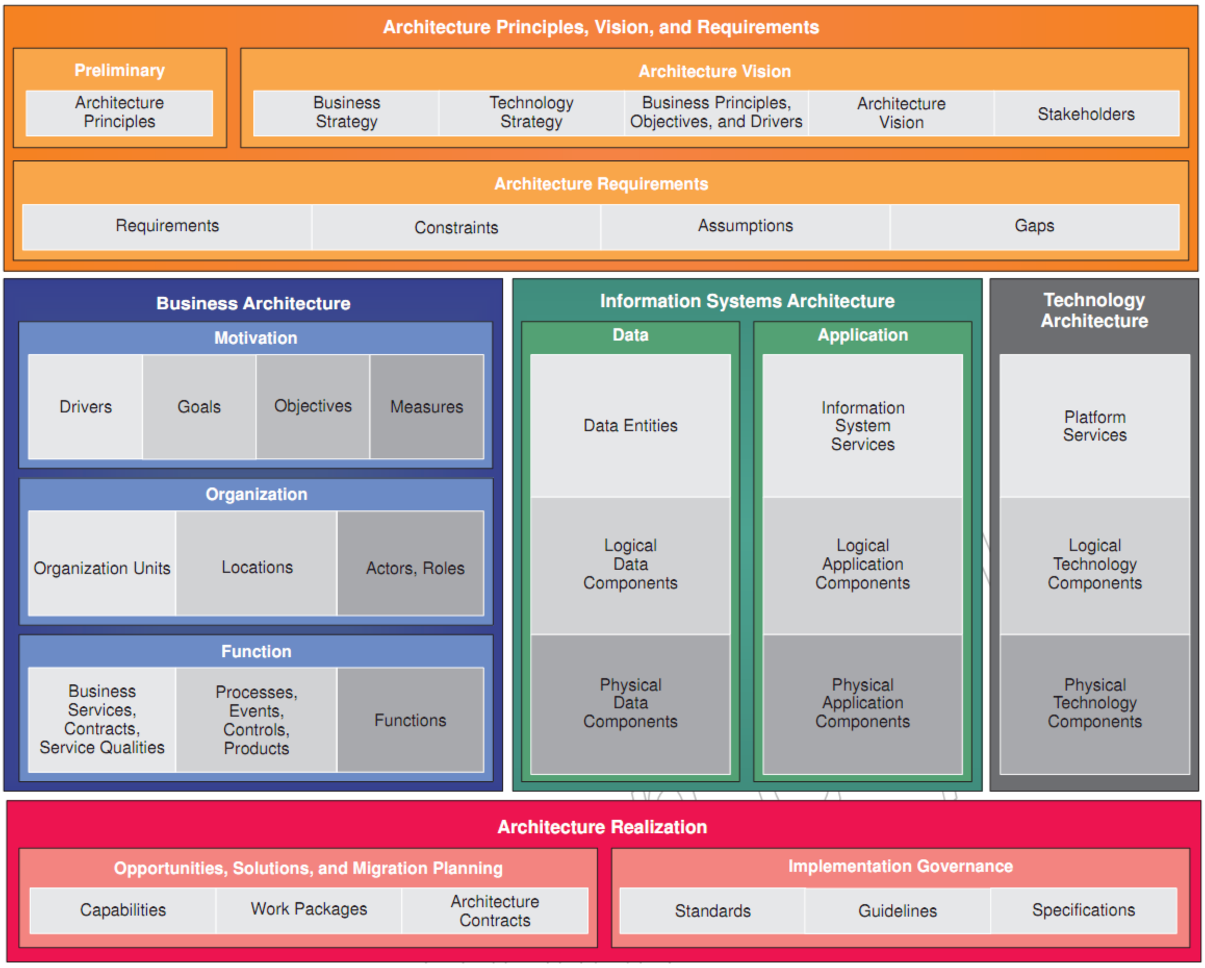


Figure 3-TOGAF Content Framework (THE OPEN GROUP, 2011)

The TOGAF Content Framework expects a number of extension modules and, as such, the TOGAF Content Metamodel was designed to support a number of extension modules that allow more in-depth consideration for particular architecture concerns. The extension modules are the *Governance Extensions*, the *Services Extensions*, the *Process Modeling Extensions*, the *Data Extensions*, the *Infrastructure Consolidation Extensions* and the *Motivation Extensions*. Figure 4 illustrates the Core Content metamodel and its predefined extension modules and their intended uses.

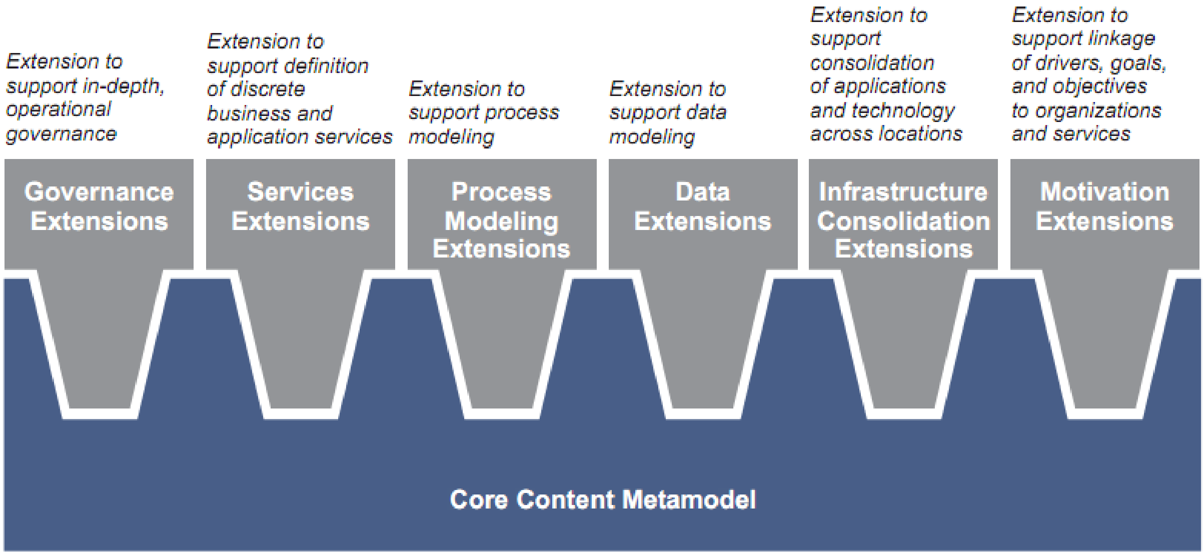


Figure 4 - Core Content Metamodel and Predefined Extension Modules (THE OPEN GROUP, 2011)

2.2.2 TOGAF Intended Motivation Extensions

As discussed in the previous section, the TOGAF Content Metamodel allows the addition of a number of predefined extension modules, including the so-called Motivation Extensions to address the motivation architectural domain. The purpose of a TOGAF motivation extension is "to allow additional structured modeling of the drivers, goals and objectives that influence an organization to provide business services to its customers". This "allows more effective definition of service contracts and better measurement of business performance".

The TOGAF specification states that the extension is intended to be used in the following situations (as extracted from (THE OPEN GROUP, 2011)):

- When the architecture needs to understand the motivation of organization in more detail than the standard business or engagement principles and objectives that are informally modeled within the core content metamodel.
- When organizations have conflicting drivers and objectives and that conflict needs to be understood and addressed in a structured form.
- When service levels are unknown or unclear.

And the expected benefits of using the extension are as follows (extracted from (THE OPEN GROUP, 2011)):

- Highlights misalignment of priorities across the enterprise and how these intersect with shared services (e.g., some organizations may be attempting to reduce costs, while others are attempting to increase capability)
- Shows competing demands for business services in a more structured fashion, allowing compromise service levels to be defined.

The TOGAF motivation extensions add elements to the TOGAF Content Metamodel and require new diagrams to be used to allow the addition of motivation extensions to the framework. As represented in Figure 5, Driver, Goal and Objective, and their respective attributes, were added as new entities that link Organization Units to Business Service. Detailed information on the full content metamodel can be found in chapter III of TOGAF specification (THE OPEN GROUP, 2011).

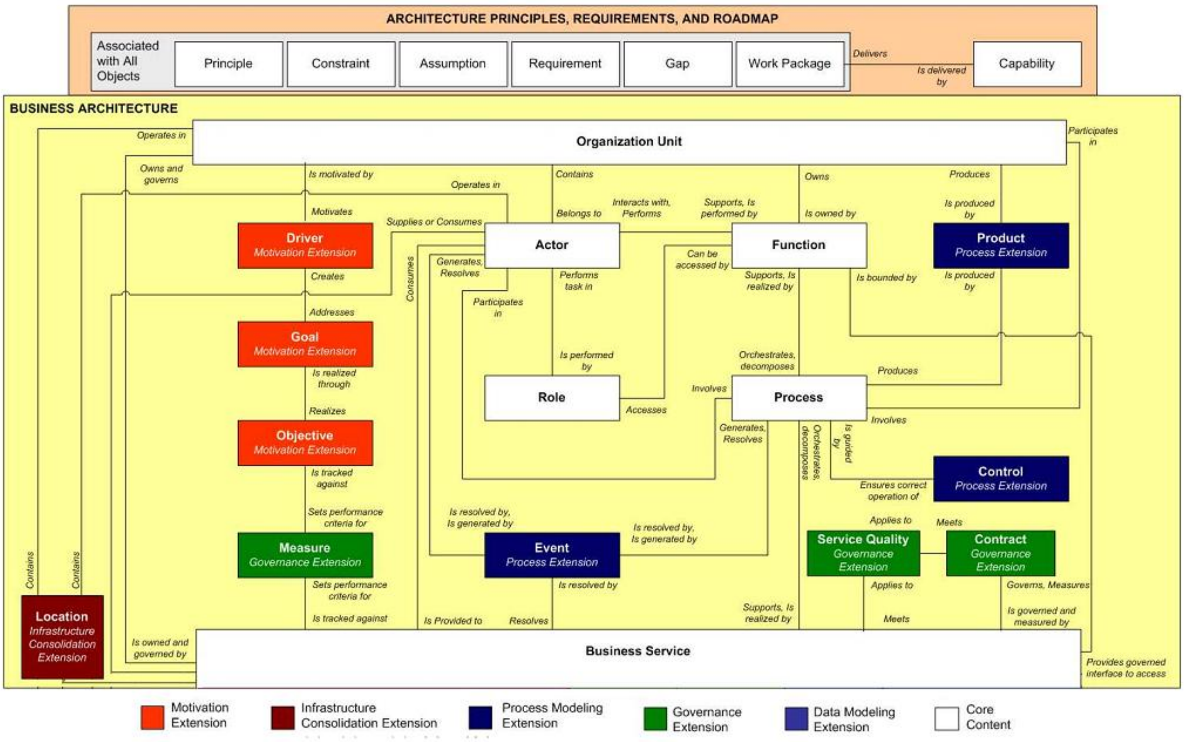


Figure 5- Fragment of the TOGAF Core Content Metamodel – adapted from (THE OPEN GROUP, 2011)

2.3 TOGAF AND ARCHIMATE

TOGAF is an architecture framework – a set of methods and tools for developing a broad range of different IT architectures. It enables IT users to design, evaluate, and build architecture for their organization, and aim at reducing the costs of planning, designing, and implementing architectures. The key to TOGAF is the Architecture Development Method (ADM) – a method for developing an IT enterprise architecture.

The ArchiMate language states that it complements TOGAF in that it provides a set of concepts, including a graphical representation, that helps to "create a consistent and integrated model "below the waterline", which can be depicted in the form of TOGAF's views". The structure of the ArchiMate language can be said to correspond with the three main architectures as addressed in the TOGAF ADM, as illustrated in Figure 6. This correspondence suggests a "fairly easy mapping between TOGAF's views and the ArchiMate viewpoints".

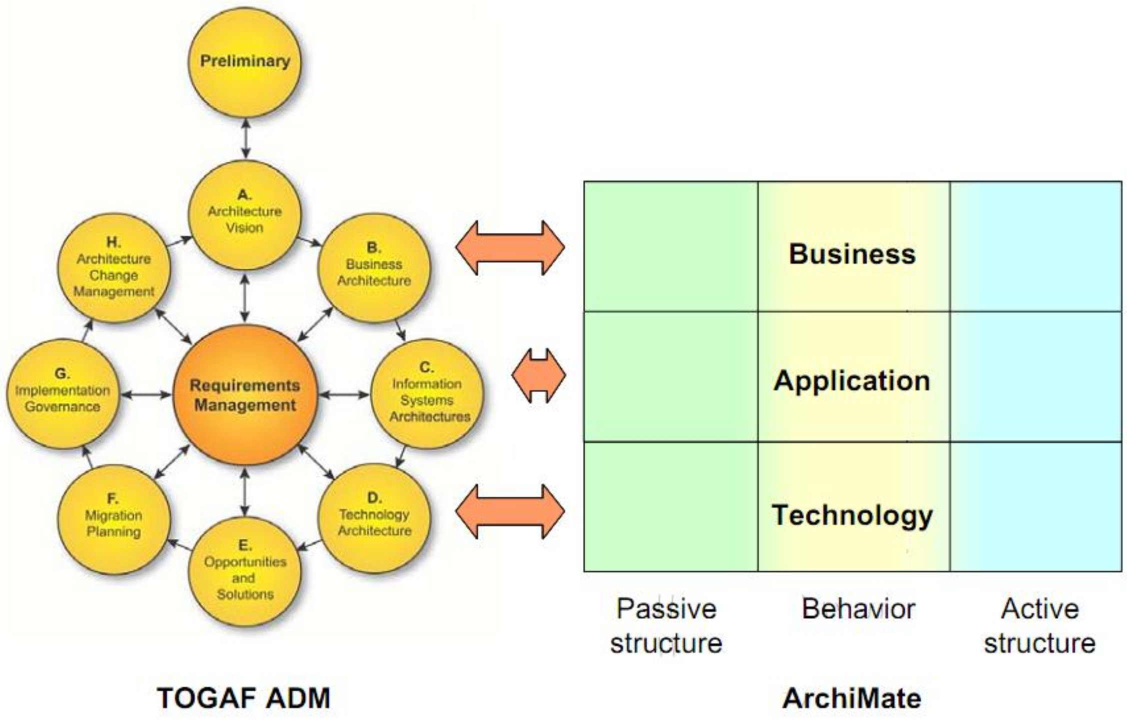


Figure 6 - Correspondences between ArchiMate and TOGAF (THE OPEN GROUP, 2009)

However, some TOGAF views are not matched in ArchiMate. TOGAF's scope is broader and addresses more of the high-level strategic issues and the lower-level engineering aspects of system development, while ArchiMate limits itself to the enterprise architecture level of abstraction and refers to other techniques for strategies, principles, and objectives, and for implementation-oriented aspects.

Although some views are not matched in ArchiMate, the language specification states that the "the ArchiMate language and its analysis techniques do support the concepts addressed in these viewpoints. Conversely, ArchiMate viewpoints that deal with the relationships between architectural layers, such as the product and application usage viewpoints, are difficult to map onto TOGAF's structure, in which views are confined to a single architectural layer".

The ArchiMate specification states that "TOGAF and ArchiMate can be used in conjunction and they appear to cover much of the same ground, be it with some differences in scope and approach".

2.4 ARCHIMATE

This section briefly discusses ArchiMate and its metamodel. For a full discussion on the topic, the reader should refer to (THE OPEN GROUP, 2009).

The ArchiMate enterprise architecture modeling language has been developed to "provide a uniform representation for architecture descriptions". ArchiMate defines architecture descriptions as "formal descriptions of an information system, organized in a way that supports reasoning about the structural and behavioral properties of the system and its evolution". The architecture descriptions define "the components or building blocks that make up the overall information system, and provide a plan from which products can be procured, and subsystems developed, that will work together to implement the overall system".

ArchiMate claims to offer an "integrated architectural approach that describes and visualizes the different architecture domains and their underlying relations and dependencies". ArchiMate has become a widely adopted language for architecture modeling in the Netherlands and is currently developed by The Open Group. (THE OPEN GROUP, 2009).

The design approach of the ArchiMate language started from a set of relatively generic concepts, which were specialized towards application on what ArchiMate names architectural layers. The language aims at simplicity and ease of use, stating that "the most important design restriction on the language is that it has been explicitly designed to be as small as possible, but still usable for most enterprise architecture modeling tasks", such that ArchiMate has been limited to the concepts that "suffice for modeling the proverbial 80% of practical cases".

The ArchiMate language consists of *active structure* elements, *behavioral* elements and *passive structure* elements. The *active structure* elements are "the business actors, application components and devices that display actual behavior, i.e., the 'subjects' of activity". The active structure concepts are assigned to behavioral concepts, to show who or what performs the behavior and those elements are named the behavioral or dynamic aspect. The *passive structure* elements are "the objects on which behavior is performed". These are "usually information or data objects in the domain of information-intensive organizations, (which is the main focus of the language)", but the *passive structure* elements may also be used to represent physical objects. ArchiMate states that the division in these three aspects – active structure, behavior, and passive structure – has been "inspired by natural language, where a sentence has a subject (active structure), a verb (behavior), and an object (passive structure)".

Figure 7 shows the so-called generic metamodel of ArchiMate and relates the core concepts to its respective aspect (shown below the metamodel).

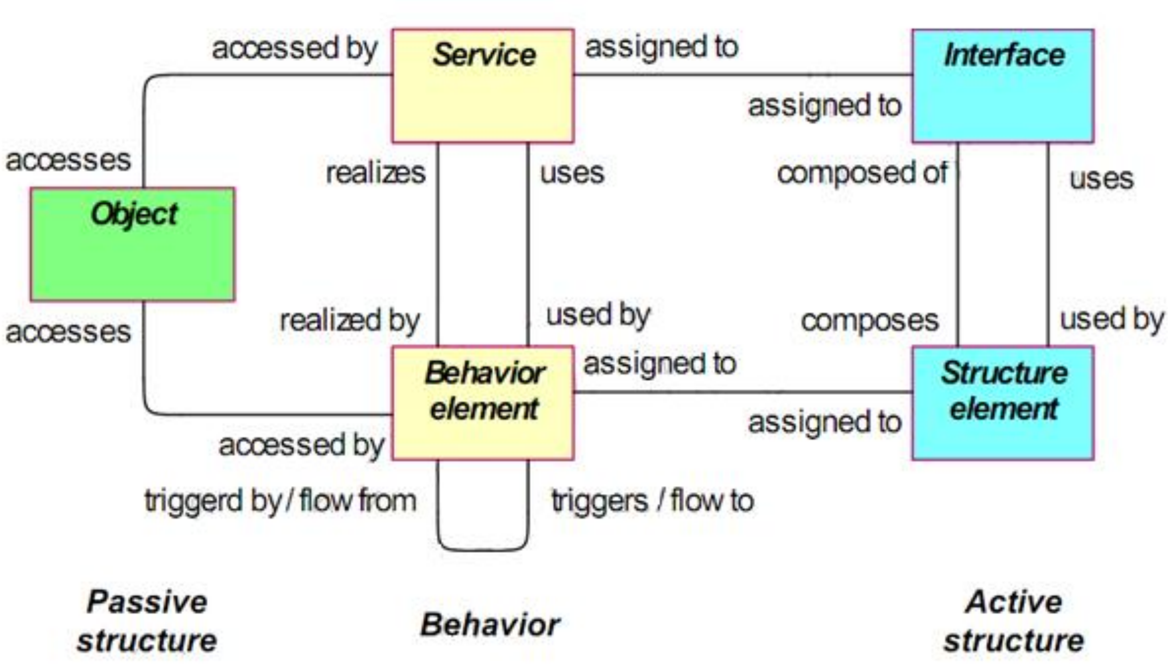


Figure 7- Generic Metamodel - The Core Concepts of ArchiMate – adapted from (THE OPEN GROUP, 2009)

The ArchiMate language defines three main layers, based on specializations of the core concepts (shown in Figure 7). These three layers (illustrated in Figure 8) are:

- The Business Layer that offers products and services to external customers, which are realized in the organization by business processes performed by business actors.
- The Application Layer that supports the business layer with application services which are realized by (software) applications
- The Technology Layer - that offers infrastructure services (e.g., processing, storage, and communication services) needed to run applications, realized by computer and communication hardware and system software.

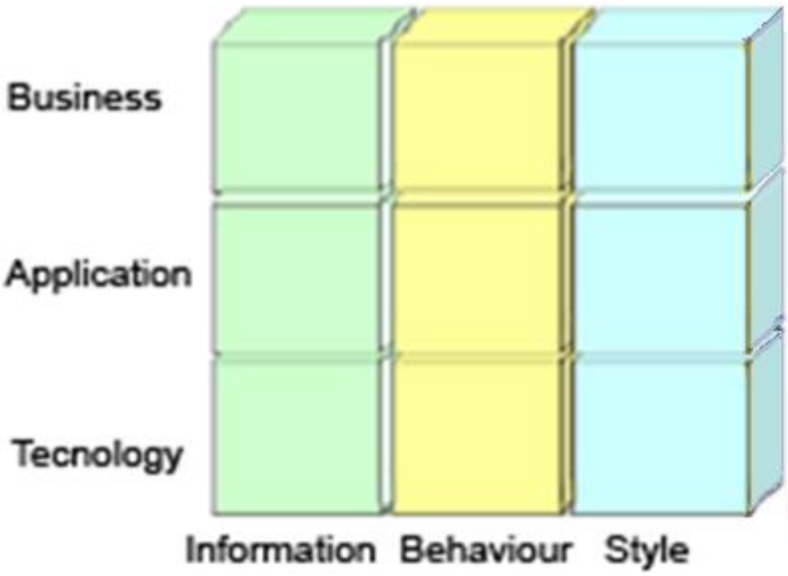


Figure 8- ArchiMate Framework

Section 2.3.1 discusses the Business Layer, section 2.3.2 discusses the application layer and the technology layer and section 2.4.3 discusses ArchiMate intended extensions and extensions mechanisms. The content shown in these sections reflects the ArchiMate specification and its descriptions and concepts were extracted from the language specification.

2.4.1 Business Layer

The structure aspect at the business layer "refers to the static structure of an organization, in terms of the entities that make up the organization and their relationships". The active entities are "the subjects (e.g., business actors or business roles) that perform behavior, such as business processes or functions (capabilities)". In the language, business actors may be individual persons (e.g., customers or employees), but may also be "groups of people (organization units) and resources that have a permanent (or at least long-term) status within the organizations".

The passive entities (business objects) are manipulated by behavior such as business processes or functions. The passive entities "represent the important concepts in which the business thinks about a domain". Figure 9 shows the business layer metamodel and Table 2 shows the description of the concepts used in the metamodel (extracted from ArchiMate specification).

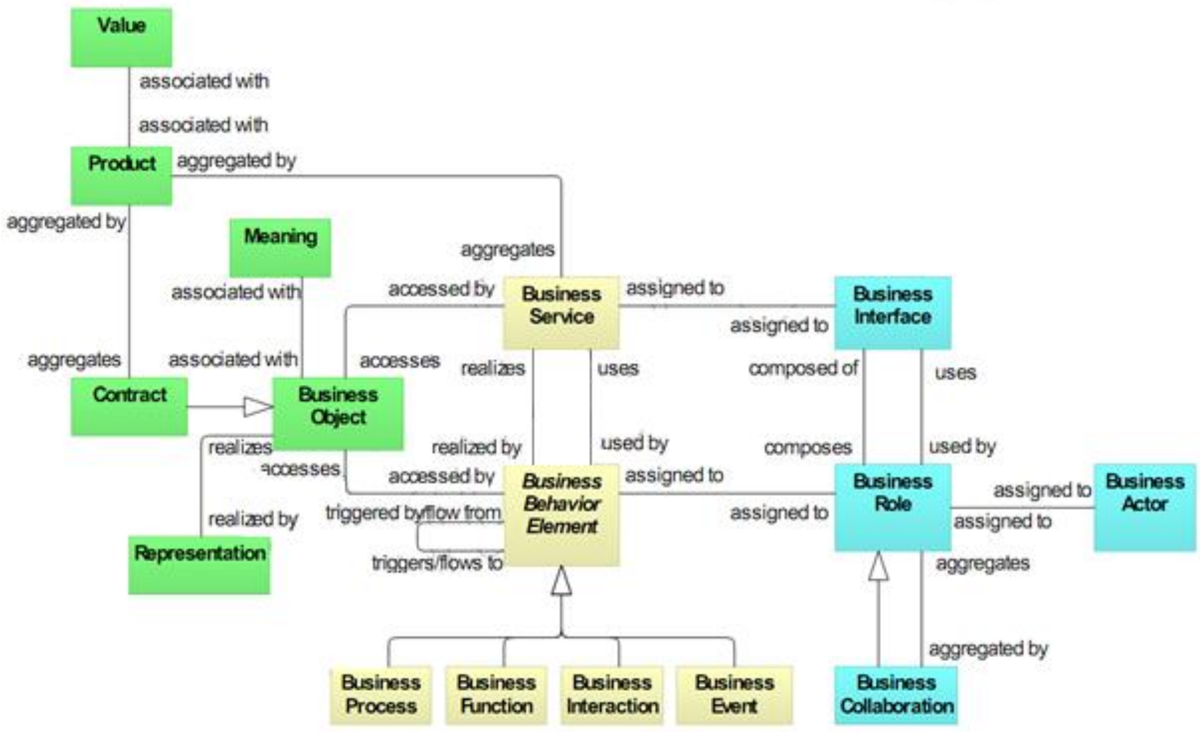


Figure 9- The Business Layer Metamodel (THE OPEN GROUP, 2009)[5]

Table 2- ArchiMate Business Application Concepts - extracted from (THE OPEN GROUP, 2009)

Concept	Description
Business actor	An organizational entity that is capable of performing behavior.
Business role	A named specific behavior of a business actor participating in a particular context.
Business collaboration	A (temporary) configuration of two or more business roles resulting in specific collective behavior in a particular context.
Business	Declares how a business role can connect with its environment.

[5] The Business Layer metamodel does not show all permitted relationships: every element in the ArchiMate language can have composition and aggregation relations with elements of the same type.

interface	
Business object	A unit of information that has relevance from a business perspective
Business process	A unit of internal behavior or collection of causally related units of internal behavior intended to produce a defined set of products and services.
Business function	A unit of internal behavior that groups behavior according to, for example, required skills, knowledge, resources, etc., and is performed by a single role within the organization
Business interaction	A unit of behavior performed as a collaboration of two or more business roles.
Business event	Something that happens (internally or externally) and influences behavior.
Business service	An externally visible unit of functionality, which is meaningful to the environment and is provided by a business role.
Representation	The perceptible form of the information carried by a business object.
Meaning	The knowledge or expertise present in the representation of a business object, given a particular context
Value	That which makes some party appreciate a service or product, possibly in relation to providing it, but more typically to acquiring it
Product	A coherent collection of services, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers

Contract	A formal or informal specification of agreement that specifies the rights and obligations associated with a product.

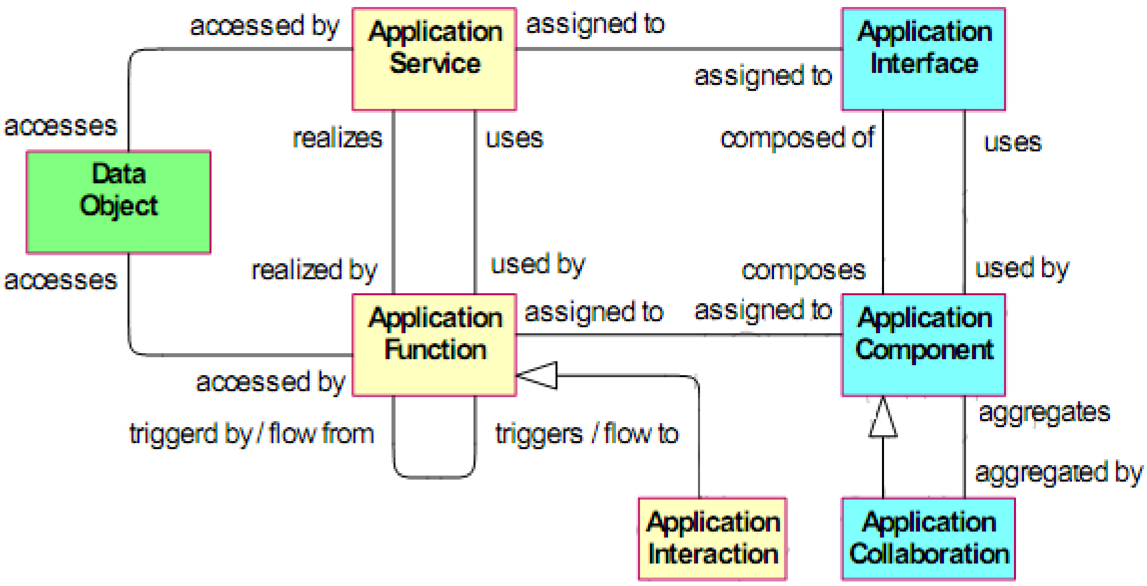
2.4.2 Application and Technology Layers

The other two layers of ArchiMate are the application layer and the technology layer.

The main structural concept for the application layer is the application component. ArchiMate uses this concept to model any structural entity in the application layer. This element can be used to model (re-usable) software components that can be part of one or more applications and, further, also (complete) software applications, sub-applications, or information systems. The application component concept strictly models structural aspects of an application: its behavior is modeled using an explicit relationship to the behavioral concepts.

The concept of application collaboration is defined in ArchiMate as a collective of application components which perform application interactions. The application interface concept can be used to model both application-to-application interfaces, which offer internal application services, and application-to business interfaces (and/or user interfaces), which offer external application service.

The passive counterpart of the component is called a data object in the language. This concept is used in the same way as data objects (or object types) in well-known data modeling approaches, most notably the "class" concept in UML class diagrams. A data object can be seen as a representation of a business object, as a counterpart of the representation concept in the business layer. Figure 10 shows the ArchiMate application layer metamodel.



[6]

Figure 10 - Application Layer Metamodel (THE OPEN GROUP, 2009)

Many of the concepts of the Technology Layer have been inspired by the UML 2.0 standard. ArchiMate states that UML is the dominant language and the *de facto* standard for describing software applications.

The main structural concept for the technology layer is the node. This concept is used to model structural entities in this layer. It is identical to the node concept of UML 2.0. An infrastructure interface is the (logical) location where the infrastructure services offered by a node can be accessed by other nodes or by application components from the application layer.

Nodes are of two types: device and system software, both taken from UML 2.0. A device models a physical computational resource, upon which artifacts may be deployed for execution. System software is classified as a behavioral concept, since it defines what a

[6] The application layer metamodel does not show all permitted relationships: every element in the ArchiMate language can have composition and aggregation relations with elements of the same type.

device "does". Typically, a node will consist of a number of sub-nodes; for example, a device such as a server and system software to model the operating system.

The inter-relationships of components in the technology layer are mainly formed by the communication infrastructure. The communication path models the relation between two or more nodes, through which these nodes can exchange information. The physical realization of a communication path is modeled with a network; i.e., a physical communication medium between two or more devices (or other networks).

An infrastructure service describes the externally visible and accessible functionality of a node. System software (similar to the "execution environment" concept of UML 2.0, but with a slightly broader interpretation) represents the software environment for specific types of components and data objects that are deployed on it in the form of artifacts. An artifact is a physical piece of information that is used or produced in a software development process, or by deployment and operation of a system. It is the representation, in the form of, for example, a file, of a data object, or an application component, and can be deployed on a node. The artifact concept has been taken from UML 2.0. Figure 11 shows the Technology Layer metamodel.

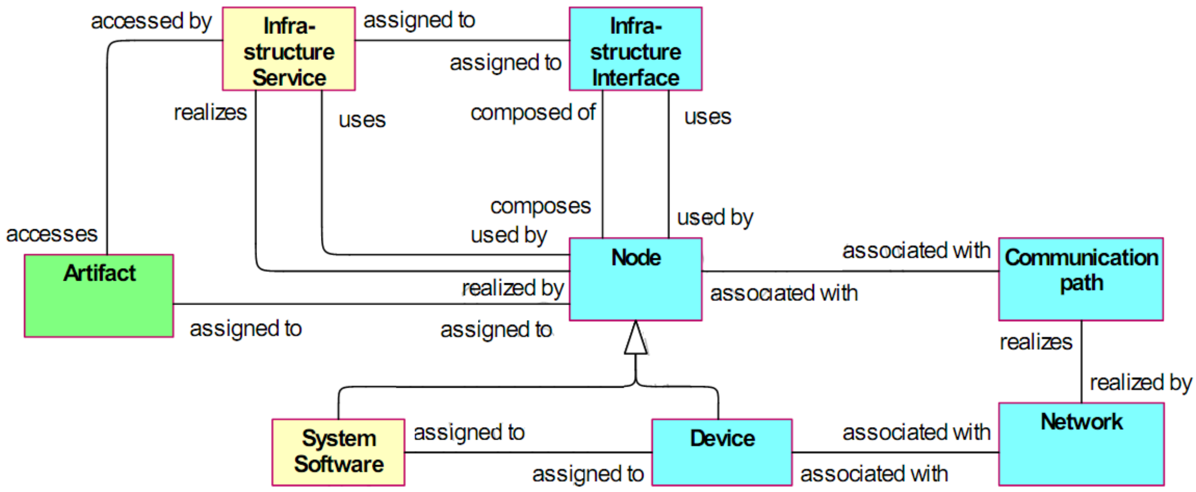


Figure 11 - Technology Layer Metamodel (THE OPEN GROUP, 2009)[8]

2.4.3 ArchiMate Extension Mechanisms and Future Directions

The ArchiMate core language, embedded in the ArchiMate metamodel, as described in sections 2.4 to section 2.4.2, claims to contain only the basic concepts and relationships that serve general enterprise architecture modeling purposes. However, the ArchiMate language also intends to facilitate extensions, through extension mechanisms. ArchiMate provides means to allow extensions of the core language that are tailored towards specific domains or applications.

The ArchiMate architecture allows basically, two types of extensions:

- Adding Attributes to ArchiMate Concepts and Relations

The core of ArchiMate contains only the concepts and relationships that are necessary for general architecture modeling. ArchiMate defines a way to enrich the language concepts and relationships in a generic way, allowing one to add supplementary information by means of a "profiling" specialization mechanism. A profile is considered a data structure

[8] The technology layer metamodel does not show all permitted relationships: every element in the ArchiMate language can have composition and aggregation relations with elements of the same type.

which can be defined separate from the ArchiMate language, but can be dynamically coupled with concepts or relationships; i.e., the user of the language is free to decide whether and when the assignment of a profile to a model element is necessary. Profiles can be specified as sets of typed attributes, by means of a profile definition language.

- Specialization of Concepts

ArchiMate states that specialization can be used as a way to define new concepts based on the existing ones. Specialized concepts inherit the properties of their "parent" concepts, but may add additional restrictions with respect to their usage. The language claims that specialization of concepts provides the needed flexibility, as it allows organizations or individual users to customize the language to their own preferences and needs, while the underlying definition of the concepts is preserved. As an example, the specification states that the concepts in the layer-specific metamodels (shown in Figure 9, Figure 10 and Figure 11) can be considered specializations of the concepts in the generic metamodel shown in Figure 7.

The current version of the ArchiMate language focuses on describing the operational aspects of an enterprise. The aim is to keep the core of the language relatively small. However, the specification states that a number of directions for extending the language are envisaged. The ArchiMate language specification has identified some likely extensions for its future versions. The language identified four fields "in which a future extension of the language may be advisable", they are:

- Strategy, goals, principles and requirements;
- Evolution and realization of architectures;
- The design process;
- Architecture-level predictions;

One of the most important fields is the Strategy, Goals, Principles and Requirements, since this future extension will relate and impact in all the other domains of the enterprise.

In the current version of ArchiMate (1.0), "the emphasis is on concepts related to the […] "extensional" aspects of the enterprise; i.e., its appearance as an operational entity".

Furthermore, "the "intentional" aspects – i.e., its business goals, principles, policies, reasons, rules, requirements, and other aspects that influence, guide, and constrain its design and operation – are less well covered". These intentional aspects would approximately correspond to the "Why" column of the Zachman framework (ZACHMAN, 1997).

ArchiMate states that the motivational domain was intentionally left out of scope in the design of the version 1.0 of the language. Indeed, ArchiMate states that an "obvious extension is the introduction of concepts to model different kinds of intentionality, both of the enterprise as a whole (e.g., business goals), and of the translation of these into restrictions on the architectural design itself (e.g., requirements, principles, rules, and constraints)". The Motivation Extension proposed to ArchiMate aims at fulfilling this gap.

CHAPTER 3. THE MOTIVATION EXTENSION TO ARCHIMATE

The Motivation Extension (ME) has been proposed in 2011 to fulfill the ArchiMate gap in the motivational domain (or intentional aspects). This extension includes a motivation column in the ArchiMate framework (see Figure 12) and proposes additional modeling concepts to address the management of goals and requirements (QUARTEL, ENGELSMAN and JONKERS, 2011).

Figure 12- Motivation Extension in the ArchiMate Framework (QUARTEL, ENGELSMAN and JONKERS, 2011)

The ME is an evolution of the ARMOR language (QUARTEL, ENGELSMAN, *et al.*, 2009), which was proposed in 2009. The design of ARMOR was driven by a rigorous analysis of existing works in the goal and requirements modeling area. For this reason, many concepts of other languages, such as i*/Tropos (BRESCIANI, PERINI, *et al.*, 2004) and Kaos (VAN LAMSWEERDE and LETIER, 2004), have influenced the development of the language. Cardoso et al. (2010) have made an analysis of the goal domain in enterprise architectures approaches and considered the ARMOR language to have a “more sophisticated support” for goal related concepts when contrasted with a number of enterprise modeling techniques. The ARMOR language has evolved into the proposed Motivation Extension for the ArchiMate framework.

3.1 THE MOTIVATION EXTENSION METAMODEL

Figure 13 shows the Motivation Extension metamodel revealing the concepts of stakeholder, concern and assessment (specializations of the 'problem element' metaclass) as well as goal, principle and requirement (specializations of the 'intention' metaclass).

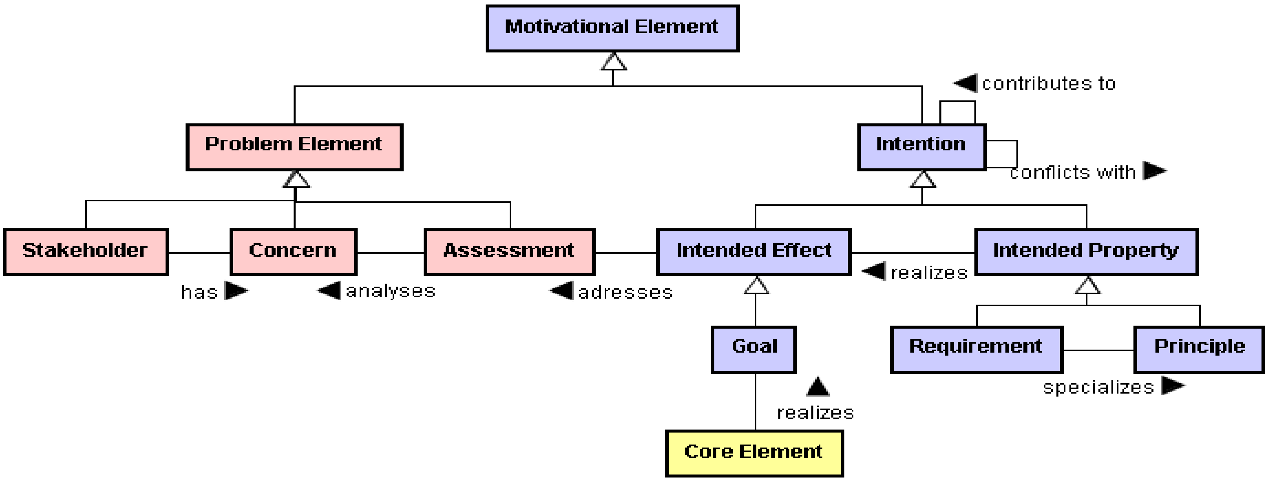


Figure 13 ME Metamodel (QUARTEL, ENGELSMAN and JONKERS, 2011)

The section 3.1.1 describes the metamodel concepts specializing Problem Element and Intention and the section 3.1.2 describes the relationships of the ME metamodel.

3.1.1 Concepts Specializing Problem Element and Intention

The ME main concepts are *Stakeholder*, *Concern*, *Assessment*, *Goal*, *Principle* and *Requirement*. Figure 14 shows the ME Notation.

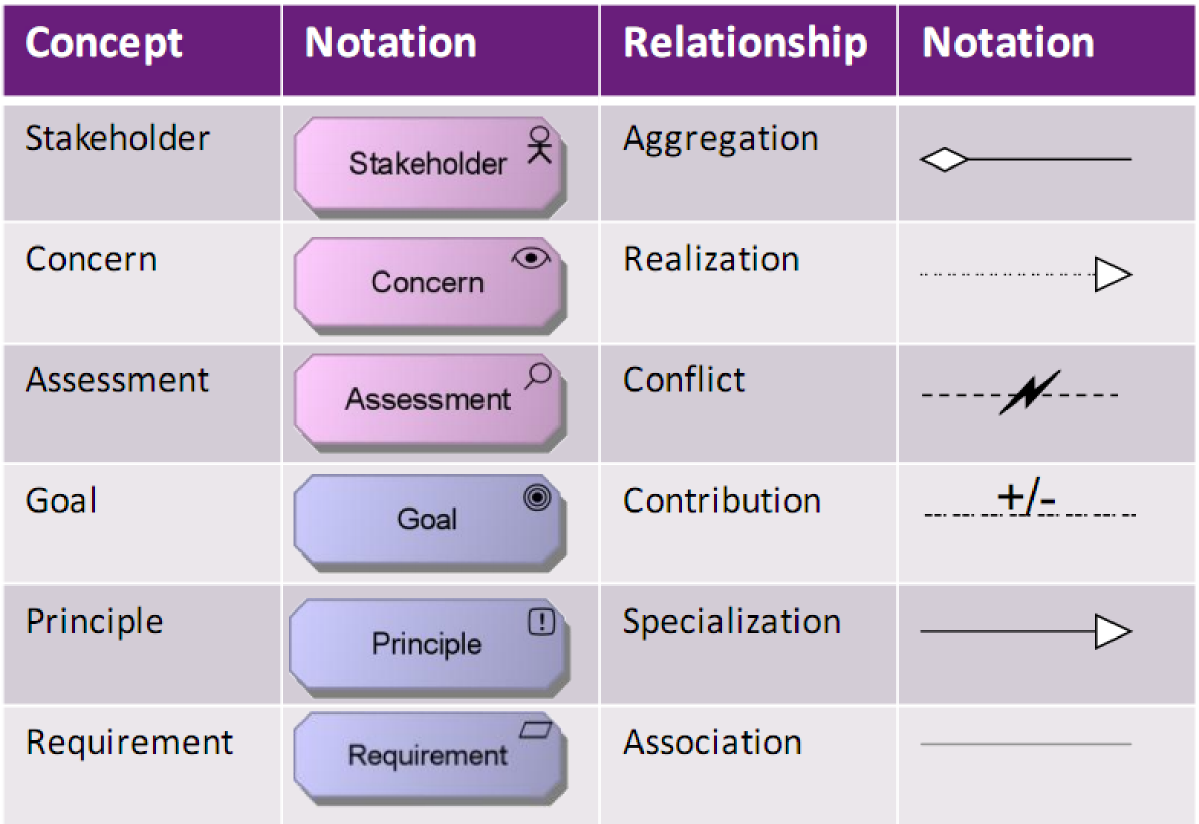


Figure 14- ME Notation (QUARTEL, ENGELSMAN and JONKERS, 2011)

3.1.1.1 Stakeholder

Intentions are pursued by people, the *stakeholders*. A stakeholder represents an "individual, team, or organization with an interest in the outcome of the architecture". The stakeholder definition is adopted from TOGAF (THE OPEN GROUP, 2009). Examples of stakeholders include the board of directors, shareholders, customers, business and application architects and legislative authorities. Figure 15 illustrates a stakeholder.

3.1.1.2 Concern

Stakeholders have certain areas of interest, named *concerns*. A concern "represents some key interest that is important to certain stakeholders in a system, and determines the acceptability of the system. A concern may pertain to any aspect of the system's functioning, development, or operation, including non-functional considerations such as performance and security". The concern definition is also adopted from TOGAF (THE OPEN GROUP, 2009). Figure 15 illustrates stakeholders' concerns.

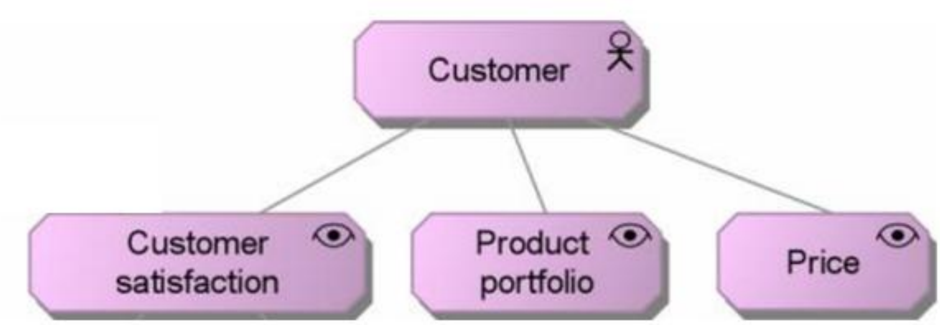


Figure 15-Stakeholder's concerns –adapted from (QUARTEL, ENGELSMAN and JONKERS, 2011)

3.1.1.3 Assessment

The *concerns* are used to organize the *stakeholders*" intentions. "*Assessments* of the *concerns* are needed to decide whether existing intentions need to be adjusted or not. The actual intentions are represented by *goals*, *principles*, and *requirements*."

An assessment "represents the outcome of the analysis of some concern, revealing the strengths, weaknesses, opportunities, or threats that may trigger a change to the enterprise architecture. These are addressed by the definition of new or adapted business goals". Figure 16 illustrates the use of assessment.

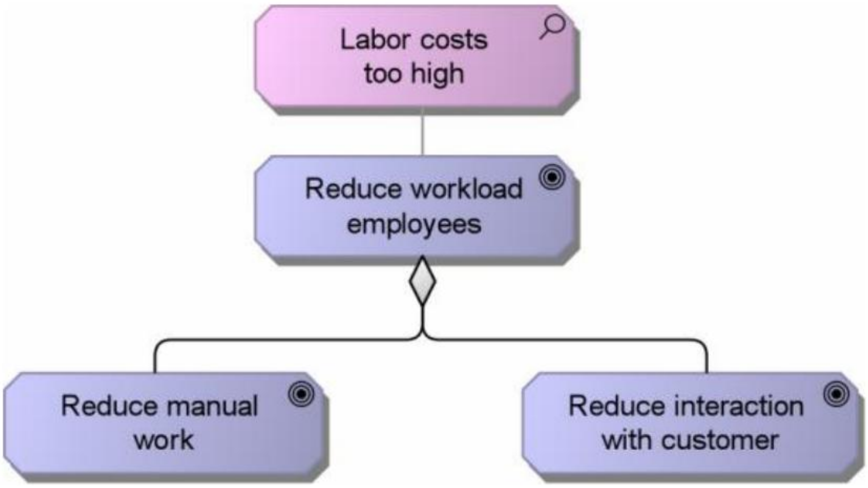


Figure 16- Assessment (QUARTEL, ENGELSMAN and JONKERS, 2011)

3.1.1.4 Goal

Goal is one of the key concepts of the language, and "a goal represents some end that a stakeholder wants to achieve." "Some end" is anything a stakeholder may desire, such as a state of affairs, a produced value or a realized effect. Examples of goals include increase profit, reduce waiting times at the helpdesk, or introduce online portfolio management. Goals can be decomposed into sub-goals. The distinction between hard goals and soft goals is implicit and the whitepaper states that "soft goals [...] need to be decomposed into hard goals". The whitepaper states that the goal "can be made measurable by adding a target or by further decomposition". For example, the Figure 17 illustrates a goal being decomposed into two, less abstract, goals.

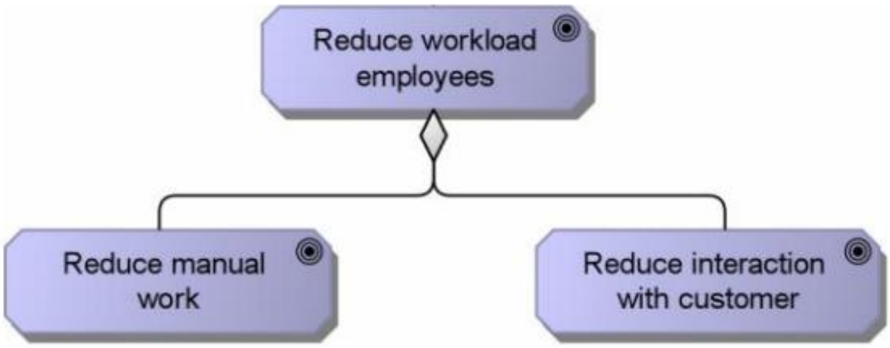


Figure 17-Soft goals and hard goals (QUARTEL, ENGELSMAN and JONKERS, 2011)

3.1.1.5 Principle

A principle "represents a general desired property that guides the design and evolution of systems in a given context. Principles are strongly related to goals and requirements. Similar to requirements, principles define desired properties of systems". The whitepaper also states that principles "are broader in scope and more abstract than requirements". A principle defines a general property that applies to any system in a certain context and a requirement defines a property that applies to a specific system. Figure 18 illustrates the use of a principle to realize goals.

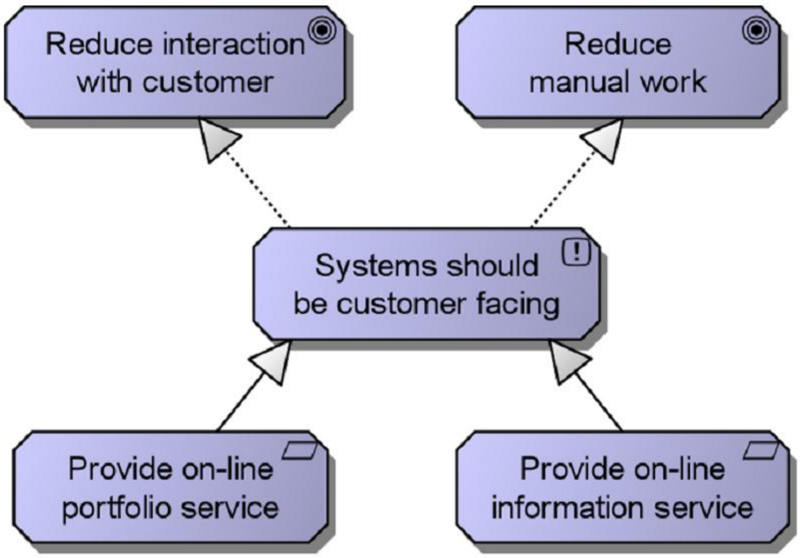


Figure 18-Goals, Principles and Requirements (QUARTEL, ENGELSMAN and JONKERS, 2011)

3.1.1.6 Requirement

A principle is motivated by a goal and "needs to be specialized into requirements in order to enforce that some system conforms to the principle". A requirement "represents a desired property that must be realized by a system". The term "system" is used in the ME in a broad sense, i.e., a system represents "a group of (functionally) related elements" and each of which may be considered a system again. A system may refer to any "structural, behavioral, or informational element of some organization". Example includes a business actor, an application component, a business process, an application service, a business object and a data object.

Requirements model the elements properties that are needed to achieve the "ends" modeled by the goals. In this respect, requirements "represent the "means" to realize goals". Goals can be "decomposed until the resulting sub-goals are sufficiently detailed to enable their realization by properties that can be exhibited by systems. At this point, goals can be realized by requirements that assign these properties to the systems". Figure 19 illustrates goals that can be realized by requirements.

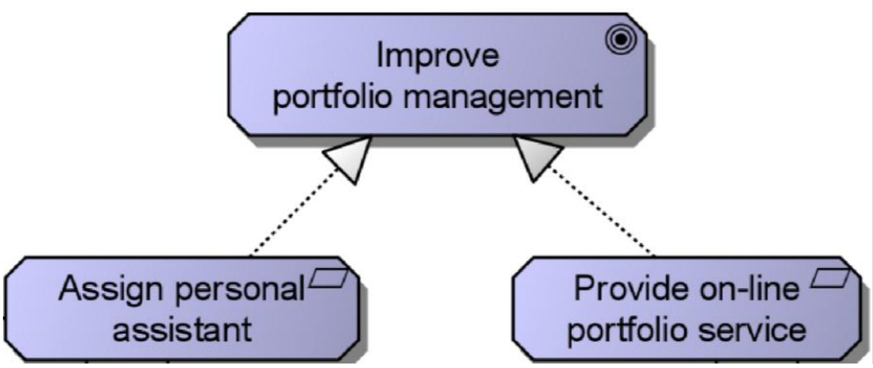


Figure 19- Goals and requirements – adapted from (QUARTEL, ENGELSMAN and JONKERS, 2011)

3.1.2 Relationships

The Motivation Extension also describes relationships between its concepts. The relationships used are the aggregation relationship, the realization relationship, the conflict relationship, the contribution relationship, the specialization relationship and the association relationship.

The aggregation relationship models "the decomposition of some intention – i.e., a goal, requirement, or principle – into more fine-grained intentions". Figure 17 illustrates the aggregation relationship.

The realization relationship "models that some end is realized by some means". Figure 19 illustrates the use of the realization relationship. The ME uses realization to describe how:

- "A goal (the end) is realized by a principle or requirement (the means)";
- "A requirement (the end) is realized by a system (the means), which can be represented by a passive structure element, a behavior element, or an active structure element";

The conflict relationship models that "the realization of two intentions mutually excludes each other". The conflict relationship is used to describe that "two intentions cannot both be realized, and as such are in conflict with each other". Figure 20 shows an example of a conflict relationship in which the goal reduce interaction with customer conflicts with the requirement to assign a personal assistant to each customer.

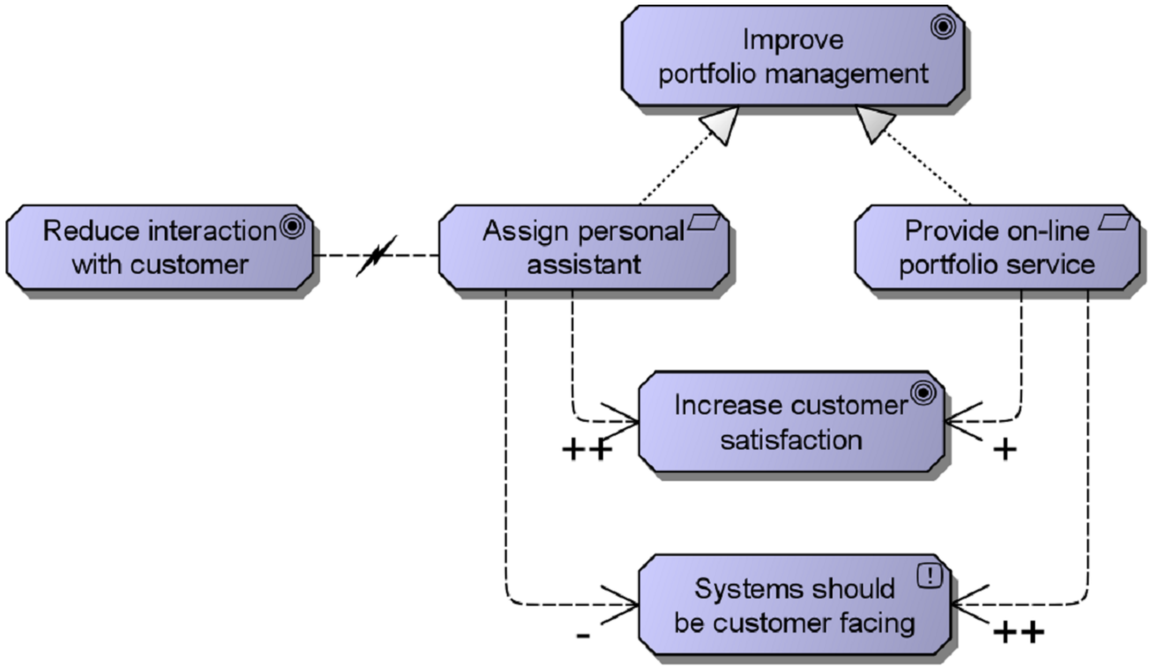


Figure 20- Contribution and Conflict relationships (QUARTEL, ENGELSMAN and JONKERS, 2011)

The contribution relationship "models that the realization of some intention contributes positively or negatively to the realization of another intention". Figure 20 depicts contributions, particularly, the contribution of the requirements to realize the goal "Improve portfolio management" to the goal "Increase customer satisfaction" and the principle "Systems should be customer-facing". The plus and minus symbols (+,-) are used as attributes to indicate the type of contribution and a scale of one or two plus or minus is provided "to indicate the [...] strength of the contribution". Also, an arrow can be used to indicate the direction of the contribution. The modeling of contribution relationships are also indicated to be used in trade-off analysis, where one is to choose between alternatives.

The specialization relationship "indicates that an object is a specialization of another object. In the current context, this relationship is used in particular to describe that a principle is specialized into a requirement". Figure 18 illustrates a specialization.

The association relationship is reused from ArchiMate to relate stakeholders to concerns and concerns to assessments and is illustrated in Figure 16.

3.2 AN EXAMPLE USING THE MOTIVATION EXTENSION

Figure 21 shows an example of an ME model (adapted from the ME whitepaper (QUARTEL, ENGELSMAN and JONKERS, 2011)). It depicts part of the stakeholders' view of an organization that sells insurances. This view shows two key characters for the organization, named stakeholders, the 'Board' and the 'Customer'. Each stakeholder has a number of concerns, in this case the Costs concern of the Board and the 'Customer satisfaction' concern, which is a shared concern between the 'Board' and the 'Customer'. Analysis of these concerns lead to assessments, in this case, the analysis of the 'Customer satisfaction' concern lead to the assessment 'Complaining customers'.

The 'Board' stakeholder also has a 'Costs' concern and its analysis leads to the assessment 'Employee costs too high'. The goal 'Reduce workload employees' addresses this assessment. This goal is decomposed into the sub-goals 'Reduce manual work' and 'Reduce interaction with customer'. The principle "Systems should be customer facing" realizes both sub-goals and is specialized into the requirements "Provide on-line portfolio service" and "Provide on-line information service"[9].

---

[9] In this section, no interpretation is given to the example. An interpretation is available in section 5.6, in which the ontology-based analysis is used to explain the intended (or unintended) semantics of the relations shown on the example.

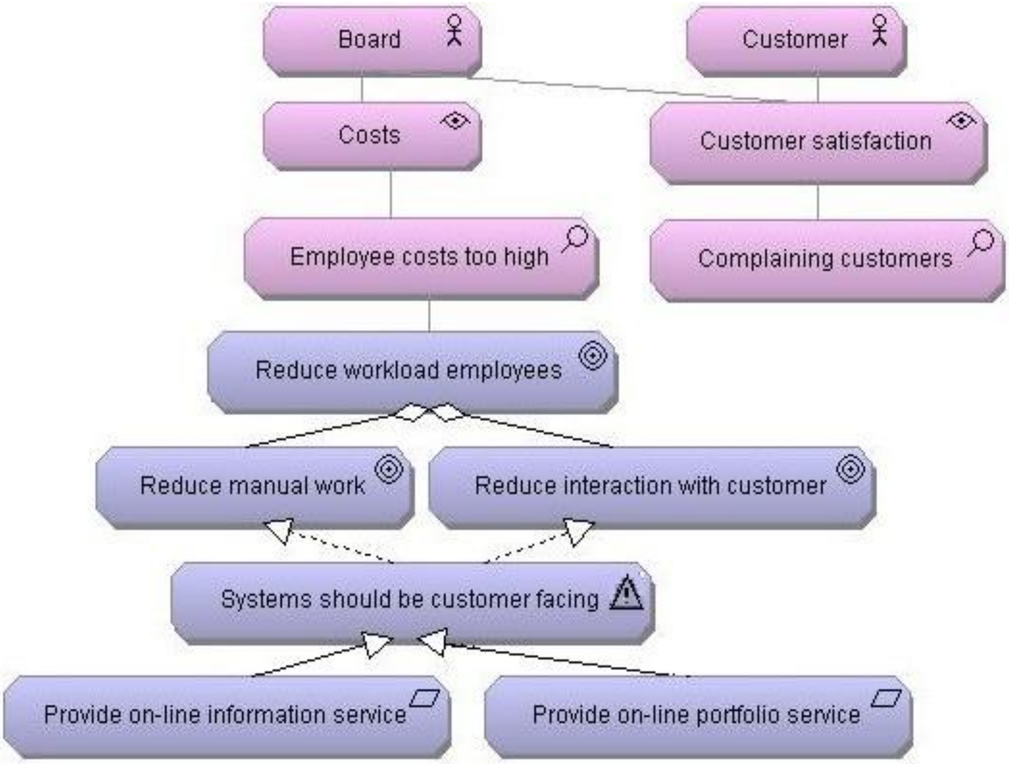


Figure 21 - ME Example

CHAPTER 4. ONTOLOGICAL FOUNDATIONS

4.1 INTRODUCTION

The term ontology comes from a Latin word which means the *study of the existence*. The term ontology have been used in philosophy both to state a discipline - with capital ‘O’ – and a language-independent category system, used to scientific theoretical conceptualizations. Ontology aims to develop theories about, for example, persistence and change, identity, causality, classification and instantiation, among others (GUIZZARDI, 2005).

The word ontology was first mentioned in a computer-related discipline in (MEALY, 1967), in a work on the foundations of data modeling. In the past few years ontologies have been a topic of interest in the computer science community. Historically, there have been three areas mainly responsible for the application of ontologies in computer science, namely, (i) database and information systems, (ii) software engineering, particularly domain engineering, and (iii) artificial intelligence;

The fields of data and information modeling applied ontological theories since the 1970’s, either implicitly or explicitly. The three-schema architecture (JARDINE, 1976) used conceptual schemas, focusing on the description of the characteristics of the existing elements in the universe of discourse.

The creation of both logical and conceptual models by the database and information modeling community was “solely motivated by the search for better concepts that could be used for creating representations of a certain portion of reality” (GUIZZARDI, 2005). The Entity-Relationship model is an example, in which these models were “committed to a world view” and that they were based on the “*ontological assumption* that the structural aspects of the world could be articulated by using the concepts of *entity* and *relationship*” (GUIZZARDI, 2005). The exemplification shows the assumption of an ontology to structure the world, even if implicitly.

The software engineering sub-field of computer science began to recognize the importance of ontologies on what came to be known as domain engineering (ARANGO, WILLIAMS and ISCOE, 1991). Domain engineering serves the purpose of a unified reference model to be used whenever ambiguities arise in discussions about the domain (*communication*) and as a source of knowledge to be used in learning processes about the referenced domain. The specification produced by the domain engineering modeling activity is "a shared representation of entities that domain experts deem relevant in a universe of discourse, which can be used to promote problem-solving, communication, learning and reuse in a higher level of abstraction" (ARANGO, 1994). Domain engineering advance was possible due to the increasing concern on the need to reduce the disproportional costs in software maintenance and also to allow software reuse in a higher level of abstraction than programming code (ARANGO, 1994). A "shared representation of entities that domain experts deem relevant in a universe of discourse" is a concern of today's domain ontologies. Indeed, domain ontologies can be used to reduce the costs in software maintenance and to allow better software reuse.

The work of Clancy (1993) was of great importance to establish the path to ontology in the artificial intelligence (AI) sub-field of computer science. In traditional AI, knowledge systems were programmed to reproduce the human experts solving steps in order to attack and solve some problem. Clancy proposed a different perspective, stating that in order to solve some problem it was needed to model the system in the world, i.e., "the primary concern of knowledge engineering is modeling systems in the world, not replicating how people think" (CLANCY, 1993). Clancy defended the modeling view of *knowledge acquisition*, in which a knowledge base is not a step by step repository of knowledge extracted from experts' minds. Indeed, this knowledge base should refer to the reality and its correspondence to the real-world. Ontologies can be used to fulfill that purpose.

Having a precise representation of a given domain is critical to fulfill the Clancy (1993) purposes, as well as to fulfill the purposes described for the three main areas that have been mainly responsible for the application of ontologies in computer science, namely, (i) database and information systems, (ii) software engineering, particularly domain engineering, and (iii) artificial intelligence. A precise representation of a given conceptualization becomes even more critical when one wants to integrate independently developed models - or systems based on those models. In order for those models or systems to function properly when used together, we need to guarantee that they ascribe compatible meanings to real-world entities of their shared subject domain.

As pointed out by (GUIZZARDI, 2005), on one hand, “a modeling language should be sufficiently expressive to suitably characterize the conceptualization of its subject domain”, on the other hand, “the semantics of the produced specifications should be clear, i.e., it should be easy for a specification designer to recognize what language constructs mean in terms of domain concepts”. Further, the specification produced using the language should “facilitate the user in understanding and reasoning about the represented state of affairs”.

The use of foundational ontologies aims to ensure ontological correctness of the language. To ensure the language correctness an ontological analysis is performed considering a mapping between language modeling constructs and the concepts in the foundational ontology. This mapping can uncover various issues related to the language ontological correctness. According to (WEBER, 1997), there should be a one-to-one correspondence between the concepts in the reference ontology and the language modeling elements. The issues that are uncovered are (illustrated in Figure 22)[10]:

- *Construct excess*

  Exists when a notation construct does not correspond to any ontological concept. Since no mapping is defined for the exceeding construct, its meaning becomes uncertain, hence, undermining the clarity of the specification. According to Weber (1997), users of a modeling language must be able to make a clear link between a modeling construct and its interpretation in terms of domain concepts. Otherwise, they will be unable to articulate precisely the meaning of the specifications they generate using the language. Therefore, a modeling language should not contain construct excess and every instance of its modeling constructs must represent an individual in the domain.

- *Construct overload*

---

[10] Extracted from (SANTOS JR, ALMEIDA e GUIZZARDI, 2012, submitted)

Exists when a single notation construct can represent multiple ontological concepts. Construct overload impacts language clarity negatively. Construct overload is considered an undesirable property of a modeling language since it causes ambiguity and, hence, undermines clarity. When a construct overload exists, users have to bring additional knowledge not contained in the specification to understand the phenomena which are being represented.

- *Construct redundancy*

  Exists when multiple modeling elements can be used to represent a single ontological concept. Construct redundancy is a violation of parsimony. In (WEBER, 1997), Weber claims that construct redundancy *"adds unnecessarily to the complexity of the modeling language"* and that *"unless users have in-depth knowledge of the grammar, they may be confused by the redundant construct. They might assume for example that the construct somehow stands for some other type of phenomenon."* Therefore, construct redundancy can also be considered to undermine representation clarity.

- *Construct deficit*

  Exists when there is no construct in the modeling language that corresponds to a particular ontological concept. Construct deficit entails lack of expressivity, i.e., that there are phenomena in the considered domain (according to a domain conceptualization) that cannot be represented by the language. Alternatively, users of the language can choose to overload an existing construct, thus, undermining clarity.

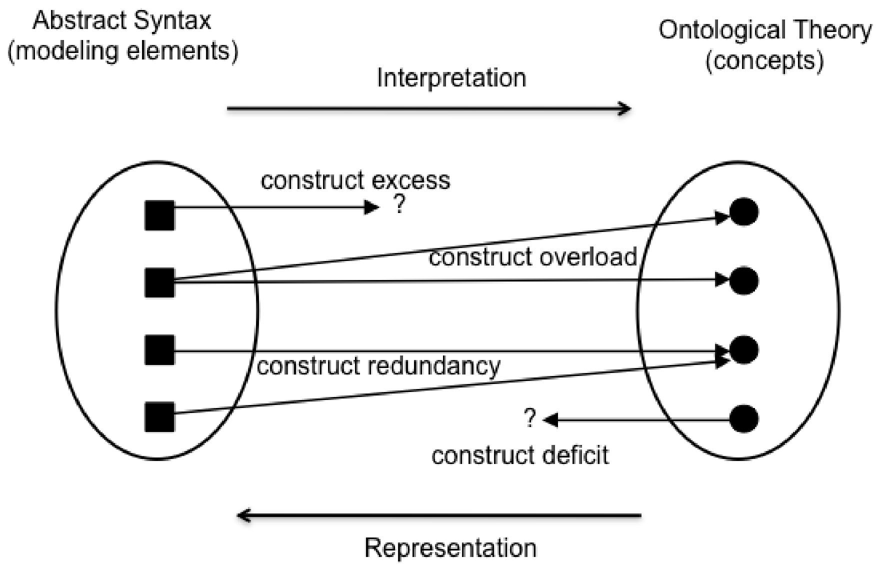


Figure 22-Issues uncovered by ontological analysis - adapted from (MOODY, 2009)

The absence of a precise definition could lead to several modeling and communication problems. For example, suppose the situation illustrated in Figure 23 of a subject domain, adapted from (GUIZZARDI, 2005). Now, suppose we have the enterprise models A and B and their respectively individual conceptualizations of the domain, the state-of-affairs they are intended to represent, namely C_A and C_B. As illustrated in Figure 23, these individual conceptualizations are not compatible nor overlap. However, because the models are based on poor representations of the domain (the model language), the models states more than just C_A or C_B, thus, their sets of possible situations overlap. As a result, models A and B agree and exchange information on situations that are neither admitted by C_A or C_B, i.e., none of them was intended to admit. Let's take the case those models are to be taken and used together, integrated. By permitting so without a precise definition, different modelers may ascribe different meanings to the same modeling elements. When sharing a model, this would lead to the False Agreement Problem (GUARINO, 1998) highlighted in Figure 23 in which each modeler would come to a different interpretation of the same model and would not detect the conflict.

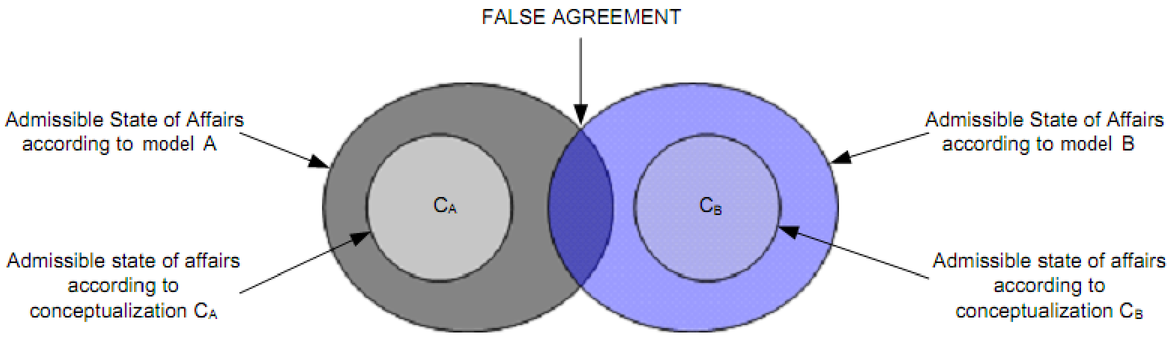


Figure 23- The False Agreement Problem- adapted from (GUIZZARDI, 2005)

This is particularly challenging in Enterprise Architecture given that an organizational model is a joint effort involving several stakeholders. Further, commonly these models requires the integration of several architectural domains into a coherent whole. The lack of common interpretation for the various modeling elements would lead to integration problems in the organization and information misuse, regardless of whether the models adhere syntactically to the same metamodel. To put it simply, although models seem to have a shared view of reality, the portions of reality that each of them aims at representing may not be compatible together. The use of expressive languages with well-defined semantics to articulate models reduces the risk of the false agreement problem.

4.2 THE UNIFIED FOUNDATIONAL ONTOLOGY

This section presents a fragment of the ontological foundation in line with the purpose of this thesis. The Unified Foundational Ontology (UFO) unifies several foundational ontologies, such as GFO/GOL (DEGEN, HELLER, *et al.*, 2001), (GUIZZARDI and WAGNER, 2005) and OntoClean/DOLCE (WELTY and GUARINO, 2001), (GUIZZARDI and WAGNER, 2005), improving some of its undesired properties on the capture of conceptual modeling basic concepts. UFO has been successfully employed to evaluate, redesign and integrate the models of conceptual modeling languages as well as to provide real-world semantics for their modeling constructs (see e.g. (SANTOS JR., ALMEIDA, *et al.*, 2010), (CARDOSO, SANTOS JR., *et al.*, 2010), (ALMEIDA, GUIZZARDI and SANTOS JR, 2009) and (GUIZZARDI and GUIZZARDI, 2011)). Further, UFO addresses social and intentional phenomena, which are modeled by the motivation domain. For a full discussion regarding this ontological foundation, refer to (GUIZZARDI, 2005) and (GUIZZARDI, FALBO and GUIZZARDI, 2008).

4.2.1 Individuals and Universals

A fundamental distinction in this ontology is between the categories of individuals[11] and universals. Universals are predicative terms that can be applied to a multitude of individuals, capturing the general aspects of such individuals. Therefore, since the categories of UFO classify these modeling elements, the fundamental category, named Thing, branches on the Universal and Individual categories.

Typically, Universals are known as a concept/class or as a relation/association, in which it describes the general concepts and properties that are applied to all the elements that are instances of that type of Universal, i.e., an Universal is a category or type that represents a pattern of properties that all of its instances necessarily have. For example, *Person*, *Dog* and *Student* are Universals, while *Carlos* is an instance of the Universal *Person*.

Individuals, on the other hand, are entities that exist in reality possessing a unique identity. Individuals are known to have unique properties, those which differentiate one individual from another. An individual must instantiate at least one Universal, but can instantiate more than one, i.e., an Individual can instantiate a multitude of universals. Examples of Individuals are Snoop that instantiates the Universal *Dog*, and *Carlos*, that instantiates the Universal *Person* and instantiates the Universal *Student*, both having its category of properties being defined by the Universals they instantiate, while having its own values for its own properties. Figure 24 shows a fragment of the UFO depicting Individuals and Universals.

[11] In some papers of the author also named as ‘Particular’

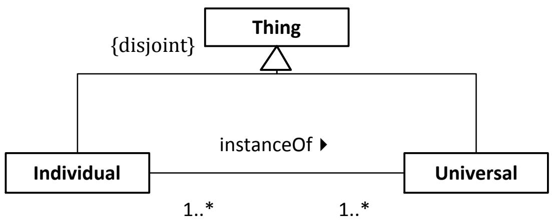


Figure 24-Fragment depicting Individuals and Universals

4.2.2 Substantials and Moments

One of the most important distinctions on UFO is on substantials and moments. Substantials and Moments apply both to Universals and Individuals, named Substantial Universal and Moment Universal, respectively. The word Moment is derived from the German Momente, from the writings of Husserl and has its origin of the notion of moment in the theory of individual accidents developed by Aristotle (GUIZZARDI, 2005). Moments are also known as 'abstract particulars' (CAMPBELL, 1990; SCHNEIDER, 2002), Tropes (SCHNEIDER, 2003) or modes (LOWE, 2001), (SCHNEIDER, 2002). An important thing that characterizes all Moments is that they are existentially dependent entities, i.e., for a moment x to exist, another individual must exist, named its bearer. Examples of moments include a color, a marriage, an electric charge on a conductor, etc. Existential dependency is a necessary but not a sufficient condition for something to be a moment. For instance, the red quantity of a color of a car depends on, but is not a moment of its color. Thus, "for an individual x to be a moment of another individual y (its bearer), a relation of inherence – sometimes called ontic predication - must hold between the two". For example, inherence glues your smile to your face, or the charge in copper to the conductor itself. Inherence is an irreflexive, asymmetric and intransitive relation between moments and other types of endurants.

Existential dependence can be used to differentiate intrinsic and relational moments. Intrinsic moments are dependent on a single individual, and are used for qualities such as weight and color, modes such as a belief, an intention, a headache, a thought, etc. Relational moments (also called relators), on the other hand, depend on a plurality of individuals. Examples include marriage, a handshake and an employment.

In contrast with moments, a substantial is an endurant that does not depend existentially on any other individual - and therefore does not inheres in another endurant. A substantial is what is usually referred by the common sense term “object”. As previously explained in this section, moments necessarily inhere in another individual. That inherence can form a chain of inherence, but always, the chain must end in a substantial. I.e., a moment must inhere in a sortal individual, a substantial. That inherence can be direct, in the case a moment inheres directly in the substantial, or indirect, the case of a moment that inheres in another moment, forming a chain, that therefore inheres in a substantial. Figure 25 illustrates substantials and moments.

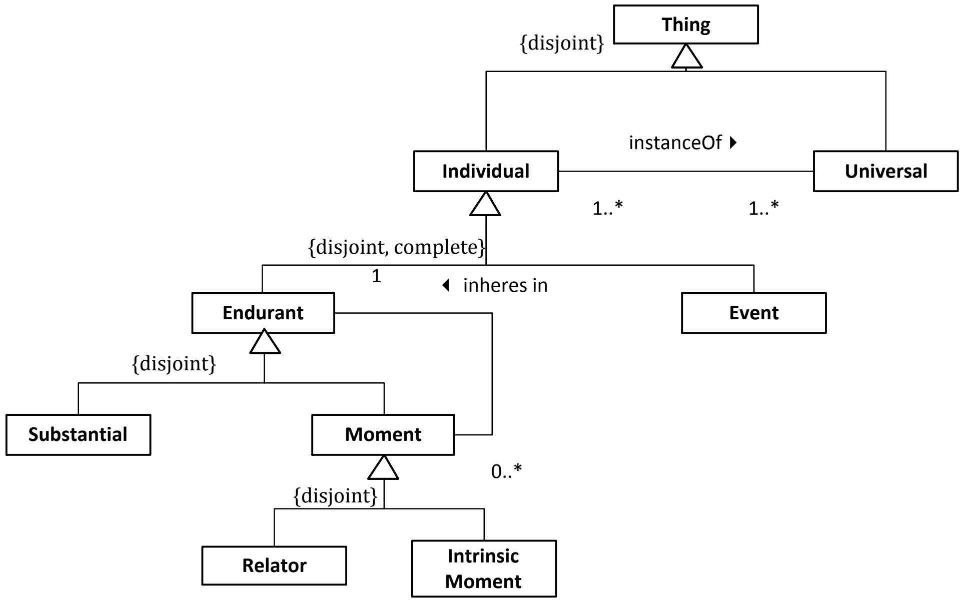


Figure 25- Substantials and Moments

4.2.3 Endurants and Events

To this work, it is important to emphasize the UFO concepts of endurants and events (also known as perdurants). The main difference between these concepts can be captured by the understanding of its behavior when related to time. Endurants are individuals that persist in time while keeping their identity, in the sense that if we say that in circumstance *c1* an endurant *e* has a property *p1* and in circumstance *c2* a property *p2* (possibly incompatible with *p1*), it is the same endurant *e* in each of these situations. Examples can include a particular person (say Peter) weighting 70kg in one circumstance and 78kg in a different circumstance, being the same individual (Peter) in all circumstances. Other examples include organizations (the University of Twente, the Federal University of Espírito Santo) and everyday objects (a ball, an apple, etc.).

Events, in contrast, are individuals composed of temporal parts. They happen in time in the sense that they extend in time accumulating temporal parts. Examples include a particular execution of a business process, a meeting, a conversation or a birthday party. Whenever an event occurs, it is not the case that all of its temporal parts also occur. For instance, if we consider a business process "Buy a Product" at different time instants, at each time instant only some of its temporal parts are occurring. As a consequence, events cannot exhibit change in time in a genuine sense since none of its temporal parts retain their identity through time.

One of the most important ontological categories for this work is an Event. "Events are possible transformations from a portion of reality to another, i.e., they may change reality by changing the state of affairs from one (pre-state) situation to another (post-state) situation. Events are ontologically dependent entities in the sense that they existentially depend on their participants in order to exist. Take for instance the event e: the stabbing of Caesar by Brutus. In this event we have the participation of Caesar himself, of Brutus and of the knife. In this case, *e* is composed of the individual participation of each of these entities."(GUIZZARDI, FALBO and GUIZZARDI, 2008) An Event can be a Complex Event or an Atomic Event, depending on its mereological parts. A Complex Event is an aggregation of at least two subparts, being each of these subparts a Complex Event or an Atomic Event. An Atomic Event is atomic, in the sense that it does not have any subparts. A Participation is an Event that existentially depends on at least two entities that participates (participants) on an event to occur. In the Caesar example, there is the participation of Caesar, the unintended participation of the knife and the participation of Brutus. Each of these participations is itself an event that can be complex or atomic, but which existentially depends on a single substantial. It is important to emphasize that the participation of two or more entities also characterizes a Complex Event. It is also important to emphasize that the classification of Atomic Events or Complex Events has no dependence on time. An Atomic Event can be instantaneous or time-extended. The same holds for Complex Events, a Complex Event can be composed of multiple instantaneous participations as well as it can be composed of time-extended participations.

As stated in (GUIZZARDI, FALBO and GUIZZARDI, 2008), "the spatial properties of events are defined in terms of the spatial properties of their participants. In contrast, all temporal properties of substantials are defined in terms of the events they participate". Guizzardi makes no strict ontological commitments on the temporal properties of these elements, in the sense that its temporal parts have their values taken by projecting these properties into some structures of a particular quality, named quality structure. The structure is composed of "Time Intervals", which are themselves composed of Time Points. These Time Intervals and Time Points can be defined by the modeler to choose on the most suitable structure to its purpose. This permits the modeling of: "(i) intervals that are delimited by begin and end points as well as open intervals; (ii) continuous and non-continuous intervals; (iii) intervals with and without duration (instants). In particular, this model allows a diversity of temporal structures such as linear, branching, parallel and circular time". For a detailed reading on the on UFO regarding the topic of qualities and its structures (including temporal structures), the reader is referred to (GUIZZARDI, 2005) and (GUIZZARDI, FALBO and GUIZZARDI, 2008). Figure 26 shows Endurants and Events.

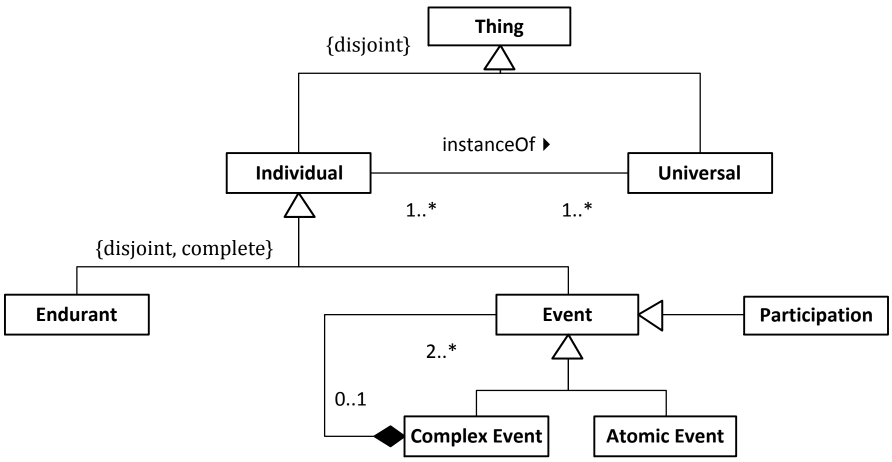


Figure 26- Endurant and Event Individuals

4.2.4 Actions

Actions are intentional events, i.e., events with the specific purpose of satisfying (the propositional content of) some intention of an agent. The propositional content of an intention is termed a goal. Only agents are said to perform actions (CARDOSO, SANTOS JR., *et al.*, 2010), as opposed to non-agentive objects which participate (non-intentionally) in events. Examples of actions include driving, writing this thesis and picking up some grocery at the supermarket. Actions, as events, can be atomic or complex. A complex action is composed of two or more participations. These participations themselves can be intentional or unintentional events. For example, writing a thesis includes the intentional participation of the author and the unintentional participation of the computer. UFO takes that it is not the case that any participation of an agent is considered an action, only those intentional participations, named action contributions. Only agents can perform actions. An object when participating in an action, is considered an unintentional participation, and is termed a resource. A complex action composed of action contributions of different agents is termed an interaction. Examples include two tennis players collaborating on a match and a dialogue between two persons. In the former case the racquetball, the ball and all the other materials are examples of resources. These object participations are named resource participation. Figure 27 depict actions.

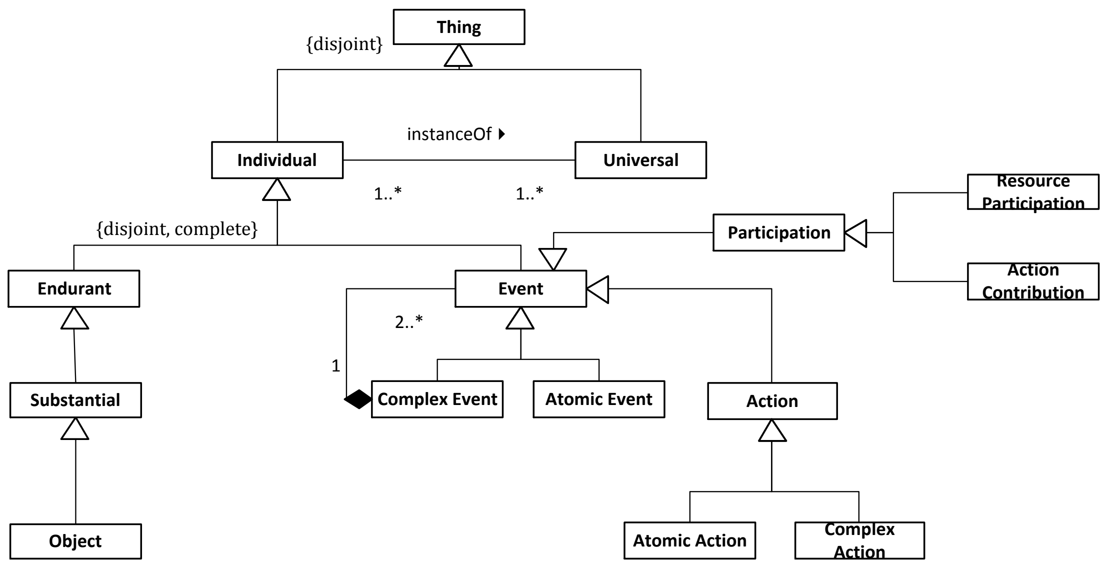


Figure 27- Actions

4.2.5 Agents and Intentional Moments

An agent is a specialization of substantial, representing entities capable of bearing intentional moments. Agents can be further specialized into physical agents (e.g., a person) and social agents (e.g., an organization). Social agents are further specialized into institutional agents and collective social agents. Institutional agents are composed of a number of other agents exemplifying what is termed a functional complex. "The parts of a functional complex have in common that they all possess a functional link with the complex. In other words, they all contribute to the functionality (or the behavior) of the complex" (GUIZZARDI, 2005). In addition to institutional agents, UFO also acknowledges the existence of collective social agents which are distinguished from institutional agents in that all its members play the same role in the collective.

Intentional moments include mental states such as individual beliefs, desires and intentions. Intentionality should not be understood as the notion of "intending something", but as the capacity to refer to possible situations of reality. This is captured in UFO with the notion that every intentional moment has an associated proposition which is called the propositional content of the moment. In general, the propositional content of an

<u>intentional</u> <u>moment</u> can be satisfied (in the logical sense) by <u>situations</u> in reality. Every <u>intentional</u> <u>moment</u> has a type (<u>belief</u>, <u>desire</u> or <u>intention</u>).

The <u>propositional</u> <u>content</u> of a <u>belief</u> is that which an agent holds as true. Examples include one's <u>belief</u> that the Eiffel Tower is in Paris and that the Earth orbits around the Sun. A <u>desire</u> expresses the will of an agent towards a state of affairs (e.g., a desire that Brazil wins the Next World Cup), while an <u>intention</u> express desired state of affairs for which the <u>agent</u> commits at pursuing (internal commitment) (e.g., my Intention of going to Paris to see the Eiffel Tower), what is commonly referred by "intention". For this reason, <u>intentions</u> cause <u>agents</u> to perform <u>actions</u>. Figure 28 shows <u>intentional</u> <u>moments</u>.

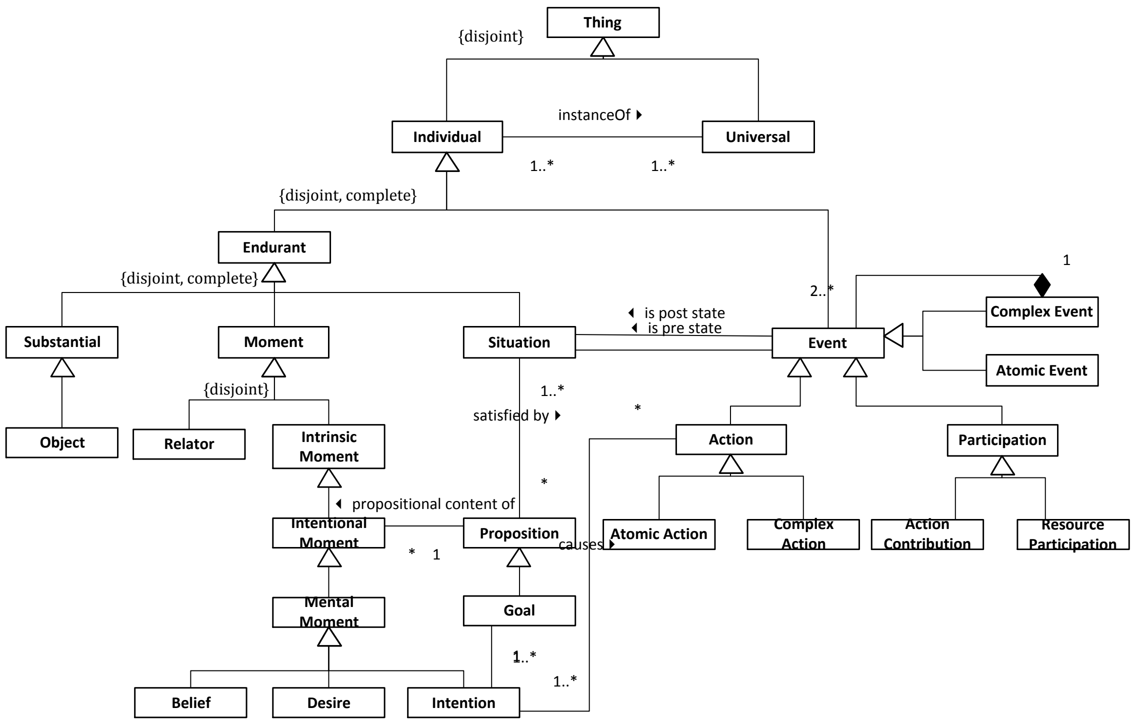


Figure 28- Intentional Moments

4.2.6 Objects

Similarly to <u>agents</u>, non-agentive <u>objects</u> can be specialized into <u>physical</u> <u>objects</u> and <u>social</u> <u>objects</u>. Physical objects include a book, a car and a paper. Social objects include money, the constitution and language. A category of <u>social</u> <u>objects</u> of particular interest to us is that of normative description. Normative descriptions are social objects that define one or more rules/norms and that can create social entities recognized in that context. Examples of normative descriptions include a company's regulations and public laws. Examples of social entities that can be defined by normative descriptions include social roles (e.g., president, manager, sales representative), social role mixins (whose instances are played by entities of different kinds, e.g., customer, which can be played by persons and organizations), social agent universals (e.g., that of political party, education institution), social agents (e.g., the Brazilian Labor Party, the University of Twente), social object universals (e.g., currency) and other social objects (e.g., the US dollar) or other normative descriptions (e.g., a piece of legislation). Normative descriptions are recognized by at least one social agent. Figure 29 shows a fragment of the specializations of individuals in UFO.

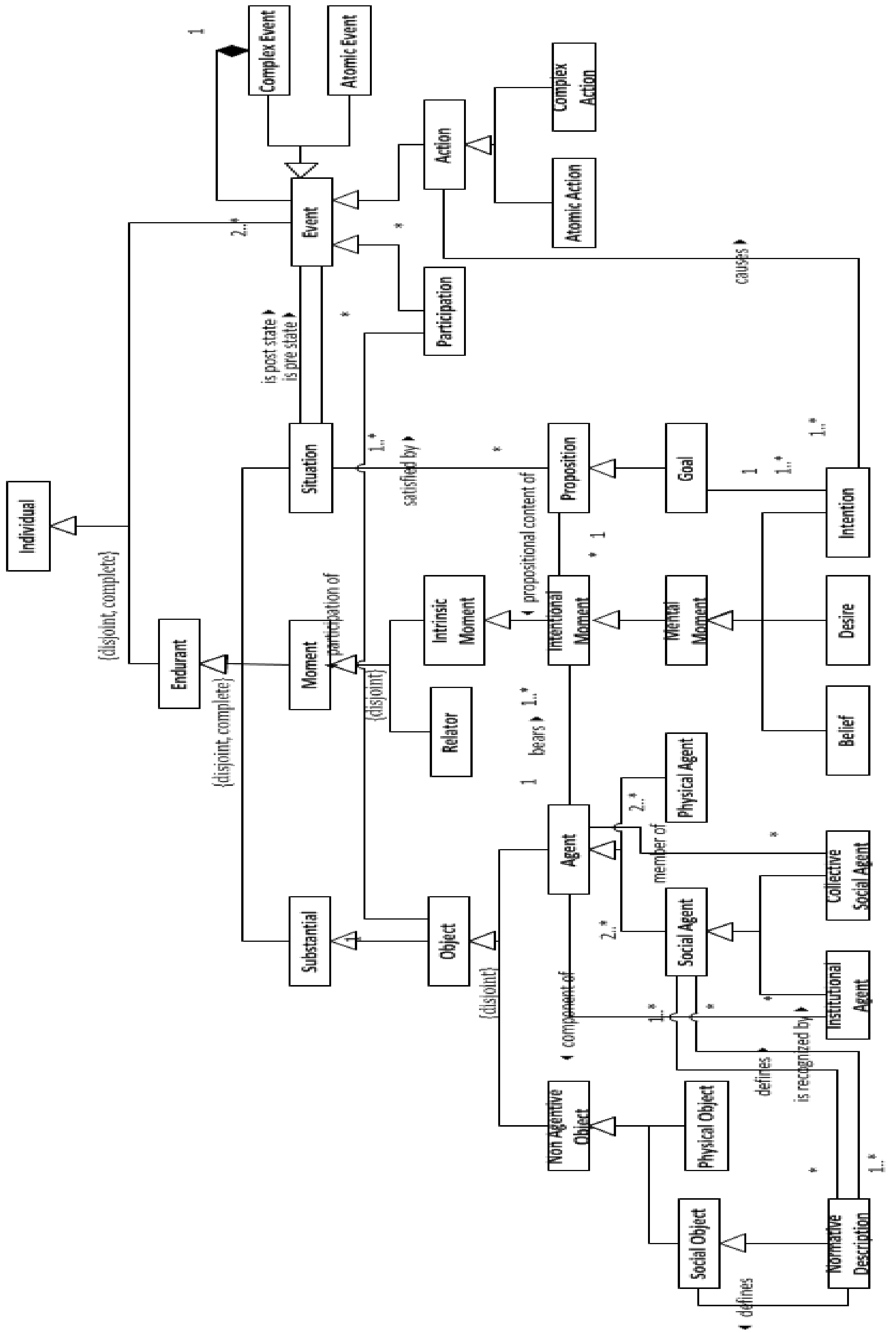


Figure 29- Specializations of individuals in UFO

CHAPTER 5. THE ONTOLOGY-BASED SEMANTICS FOR THE MOTIVATION EXTENSION TO ARCHIMATE

5.1 INTRODUCTION

This chapter presents the Motivation Extension ontological interpretation. Using as a starting point the original ME definitions from (QUARTEL, ENGELSMAN and JONKERS, 2011), as discussed in CHAPTER 4, each ME concept is analyzed and its possible ontological interpretations are discussed.

The aim is to analyze the language, and, in case of ontological deficiencies are revealed, propose a path to a solution[12].

This chapter is structured in the following sub-sections: sub-section 5.2 addresses the specializations of the ME's abstract metaclass Problem Element; sub-section 5.3 addresses the specializations of the ME's abstract metaclass Intention; sub-section 5.4 analyzes the relationships that are common to all concepts. A summary of the interpretation results is provided in subsection 5.5. An example is presented in subsection 5.6 and related work is discussed in subsection 5.7.

5.2 ANALYSIS OF SPECIALIZATIONS OF PROBLEM ELEMENT

This subsection analyzes the concepts that specialize *Problem Element*, namely *Stakeholder*, *Concern* and *Assessment*.

5.2.1 Stakeholder

The *Stakeholder* concept in the ME has been borrowed from the TOGAF framework (THE OPEN GROUP, 2009). TOGAF defines stakeholder as "an individual, team, or organization

---

[12] Results of the work shown in this chapter were presented at (AZEVEDO, ALMEIDA, *et al.*, 2011)

(or classes thereof) with interests in, or concerns relative to, the outcome of the architecture". A stakeholder is an entity able to refer to the reality (in this case "the architecture").

The *stakeholder* concept is interpreted as an Agent in UFO, a substantial that bears intentional moments (such as beliefs, desires and intentions) that refer to the enterprise architecture's elements. This interpretation also addresses the case in which a *stakeholder* is a team or a group. This is based on the fact that agents are specialized by Institutional Agents and Collective Agents, that can represent teams and groups.

The fragment "or classes thereof" in the definition of stakeholder must be carefully treated when semantics are to be given. This fragment points to the possibility of modeling classes of individuals. These classes of individuals are to be interpreted as some kind of universal in UFO that can be instantiated by agents.

These possible semantic interpretations to the same concept represent a problem known as construct overload. In this case, it indicates the need to specialize the *stakeholder* concept into at least two different elements, one to represent an agent and the other to represent a universal that can be instantiated by agents. Further, this problem would trigger other similar problems, since agents can be used to instantiate different sorts of universals, namely, Agent Universals (e.g. 'Person', 'Non-Governmental Organization'), Social Roles (e.g., 'Husband', 'Wife', 'Insurer', 'Insured') and Social Role Mixins (e.g., 'Customer'). Within the first *stakeholder* specialization, the language would then collapse the different kinds of universals, again suggesting new specializations of the *stakeholder* concept.

5.2.2 Concern

Now we focus on the concept of *Concern* and the relation *Stakeholder has Concern*. The description of the concept in the ME is abstract and states that a concern is "some key interest that is important to certain stakeholders in a system". The *concern* itself, i.e., that which is important, is only represented in the ME as the label of the modeling concept. The properties or the characteristics which are supposed to be the focus of attention are not explicitly represented in the model.

The presence of the stakeholder has concern pattern in the model indicates that the *stakeholder* (an Agent) holds a particular Belief. In UFO, every Belief entails a Proposition which is the content of the belief, i.e., that which is held as true. The proposition in the case of a *stakeholder's* concern refers to the importance that the *stakeholder* ascribes to certain *concern*. Since the properties or characteristics which are supposed to be the focus of the *concern's* attention are not explicitly represented in the language, at this moment a detailed interpretation of the possible contents of the proposition associated with a *stakeholder's concern* is out of the scope of this thesis. Nevertheless, we can state that, when a particular *concern* is associated with multiple *stakeholders*, the object of the *concern*, its proposition, is shared between all the referenced *stakeholders*, while the belief itself is particular to an individual *stakeholder*.

5.2.3 Assessment

The *assessment* concept "represents the outcome of the analysis of some *concern*". It indicates that some agent makes an analysis about a *concern*, drawing conclusions from this analysis. Semantically, in UFO, this means that some agent acquires new beliefs with propositional contents referring to a specific situation. The propositional content of the belief refers specifically to the properties or characteristics believed to be important (the object of the *concern*). This interpretation refers both to the *assessment* itself and the *assessment analyzes concern* relation, since an *assessment*, conceptually, needs to be about something. The ME does not indicate who performs the *assessment*. Thus, while the propositional content of the belief is settled (informally in the assessment's label), the identity of the agent remains undetermined. Therefore, we interpret an *assessment* as a proposition of the belief of some undetermined agent (or agents).

This gap of knowledge on who performs the assessment shows that the ME might lack a construct to relate *assessments* and *stakeholders* - to determine the identity of the agent that holds the belief. This is particularly important as different *stakeholders* may come to different *assessments* of a shared *concern*. In the example in section 4.2, both the "Board" and the "Client" are concerned with "Customer Satisfaction"; consider the case in which the analysis of "Customer Satisfaction" by the "Board" lead to different conclusions when contrasted with the analysis of the *concern* by the "Client" (consider a situation in which the

"Board" assesses that "Customer Satisfaction is High", while the "Client' assess that "Customer Satisfaction is Low"). This situation would be perceived differently by the two agents, which would hold beliefs with diverging propositional contents. This situation is not satisfactorily addressed by the ME.

For the cases in which a *stakeholder* is interpreted as an Agent Universal, it is understood in this thesis that each instance of this universal has a belief with that propositional content, i.e., all *stakeholder* instances share the *assessment*.

5.3 ANALYSIS OF SPECIALIZATIONS OF INTENTION

This subsection focuses on the concepts that specialize *Intention*, namely, *Goals*, *Principles* and *Requirements*.

5.3.1 Goal

A *goal* in the ME "represents some end that a stakeholder wants to achieve", which can be a "produced value or a realized effect". From the ME definition, we can observe that:

(i) A stakeholder is committed to achieving a goal; and that,
(ii) achieving the goal means bringing out certain effects in reality.

Semantically, in UFO, this means that some agent has the intention of bringing about the goal. Thus, the agent intends to perform actions that have as post-state a situation (a state-of-affairs) that satisfies the goal. Since *goals* in the ME can be a "produced value or a realized effect", the situations that satisfies the goal are the ones in which this value has been produced or this effect is realized. Similarly to the case of assessment, the ME does not indicate the *stakeholder* who has the *goal*. Thus, while the propositional content of the intention is settled (informally in the goal's label), the identity of the agent remains undetermined. Therefore, we interpret a *goal* as a goal (proposition) of some undetermined agent (or agents).

The ME might lack a construct to relate *goals* and *stakeholders*, as shown above, to determine the identity of the agent that holds the intention of bringing about the goal. A possible solution is the addition of a constructor with this function to the language.

In the ME, *goals* are defined in order to address *assessments*. The extension includes a *Goal addresses Assessment* relationship to capture this notion. Informally, the *goal* is motivated by the wish to change the situation that is revealed by the *assessment*. This assessment reveals that a situation is not desired and some actions are to be taken in order to change this state of affairs. Semantically, this wish can be understood as an implicit desire that is not satisfied in the situation revealed by the *assessment*. The agent then adopts a goal, committing to pursue the desired situation.

5.3.2 Principle

The *Principle* concept "represents a general desired property" that applies to any system in a given context. The idea is that the organization wants its systems (for the ME, a system is "a group of (functionally) related elements, each of which may be considered as a system again. A system may refer to any structural, behavioral or informational element of some organization, such as a business actor, application component, business process, application service, business object, data object, etc.") to show specific properties. A *principle* would ideally impact in the design or in the actions of the systems.

The *principle* concept is interpreted in UFO as a desire. A desire in UFO has a propositional content and also means that there is not necessarily an agent which has a commitment (self or social commitment) to act now on the desire to make it become true. The propositional content P of the desire that corresponds to the *principle* is the result of the application of the predicate Q on all systems in a given context, i.e., $P \equiv \forall s((System(s) \wedge ContextPrinciple(s)) \rightarrow Q(s))$, where System holds for all systems, ContextPrinciple holds for all systems in the context of application of the *principle* and Q holds for the systems that exhibit the desired properties stated in the *principle*.

The ME does not relate *principles* and *stakeholders*, thus it is understood that the agent that is implied in a *principle* is the enterprise as a whole. If this is not the case, then the language should include additional relations to clarify the specific *stakeholders* that hold the *principle*.

This may be relevant when conflicting *principles* are considered for different *stakeholders* in an intra-enterprise or inter-enterprise setting.

5.3.3 Requirement

The *Requirement* concept is actually similar to the *principle* concept. A *principle* represents a general **desired** property that applies to any system in a context, while a *requirement* "represents a desired property that **must** be realized by a [specific] system". Similarly to *principle*, the definition of *requirement* seems to imply that a desire for specific properties exist, justifying an interpretation of *requirement* as a Desire in UFO. However, differently from *principle*, *requirements* must be satisfied if the system is to exist. This suggests an additional normative character of a *requirement*, which justifies an interpretation of a *requirement* as a normative description. The propositional content of the desire (representing the *requirement*) is formulated as a normative description which states that if a system is to exist, then it must satisfy the *requirement's* proposition. The propositional content R of the desire that corresponds to the *requirement* is the application of the predicate Q on a specific system (to-be) A, i.e., $R \equiv Q(A)$. Ideally, the desire would lead to the adoption of intentions to satisfy the *requirement*, for example, by committing to actions to develop systems that satisfy the *requirement*. Nevertheless, this may not be the case for *requirements* with a low priority or *requirements* whose realization may not be worthwhile (in which case the *stakeholder* will not commit to the actions that pursue *requirements* satisfaction).

5.3.4 Relations Involving Intention and its Specializations

Now we focus on the relations involving *Intention* and its specializations (*Principles*, *Requirements* and *Goals*). We start with the *realization* relationship that "models that some end is realized by some means." The *realization* relationship is applied in three different relations, and have different, but compliant interpretations for each, as follows:

(i) *Core element realizes Requirement*. The presence of this relationship indicates that the *core element* exhibits the *required* properties. In other words, the propositional content of the desire (stated as a normative description corresponding to the *requirement* R) is satisfied in all the situations which include the stated *core element*. Since, R is a predicate on the stated *core element*, R holds.

(ii) *Requirement realizes Goal*. The presence of this relationship indicates that the goal (proposition) G is satisfied in all situations that satisfy the propositional content R in the *requirement*. This relation reflects a logical relation between R and G, namely, $R \rightarrow G$.

(iii) *Principle realizes Goal*. The presence of this relationship indicates that the goal G is satisfied in all situations that satisfy the propositional content P of the *principle*. This relation also reflects a logical relation between P and G, namely, $P \rightarrow G$.

We should now refer to the *specialization* relation between *Principle* and *Requirement*. The current documentation of the ME states that "a principle needs to be specialized into requirements in order to enforce that some system conforms to the principle." Is understood that the presence of this relationship indicates that the system that satisfies the *requirement* exhibit the properties stated in the *principle*. Reminding that the propositional content P of the principle has the following structure:

$P \equiv \forall s((System(s) \wedge ContextPrinciple(s)) \rightarrow Q(s))$, where Q holds for the systems that exhibit the desired properties stated in the principle.

Thus, formally, the presence of the *specializes* relationship indicates that the predicate Q (properties stated in the *principle*) is satisfied in all situations that satisfy the propositional content R in the *requirement*. Since R is a proposition that states properties of a particular system (call this system S), then this relation reflects a logical relation between R and Q(S), namely, $R \rightarrow Q(S)$. In other words, the satisfaction of the *requirement* (in S) satisfies the principle's properties when applied to S.

The *conflict* relationship "is used to describe that two intentions cannot be realized both, and as such are in conflict with each other". In other words, one *intention* leads the world to a state in which the other *intention* cannot be satisfied. In UFO, we can say that from the current situation there are no events (or complex sequences thereof) that would lead to a situation in which the propositional content of both *intentions*

(*Principle*/*Goal*/*Requirement*) would hold. Note that this is different from the case in which the propositional contents of the *intentions* are mutually incompatible, i.e., cannot hold simultaneously.

According to this interpretation, *intentions* with incompatible propositions are also in conflict with each other. Nevertheless, it is important to remind that there are *intentions*, which, while in principle compatible with each other, cannot be accomplished together from the current situation. For example, an organization holds U$100.000,00 for investments and intends to acquire two machines that are worth U$100.000,00 each. While the propositional contents are not mutually incompatible there are no events (or complex sequences thereof) that would lead to the accomplishment of both *intentions*.

The *contribution* relationship models that "the realization of some intention contributes positively or negatively to the realization of another intention". In a positive contribution, less effort is required to reach a situation that satisfies the propositional content of B from the situation that satisfies the propositional content of A. In a negative contribution, more effort is required to reach a situation that satisfies the propositional content of B from the situation that satisfies the propositional content of A.

5.4 COMMON RELATIONSHIPS ANALYSIS

This section analyzes the relationships that are common for all the concepts described in ArchiMate and have no specific definition in the ME, namely *aggregation* and *specialization*.

According to the ME whitepaper, "the aggregation relationship models the decomposition of some intention, i.e., a goal, requirement or principle, into more fine-grained intentions." While this definition only discusses explicitly the metaclasses which specialize intention, according to ArchiMate, "the aggregation relationship can relate any instance of a concept with another instance of the same concept". Thus, the conclusion is that the relationship can also be used to decompose *assessments*, *concerns* and *stakeholders*.

At first observation, the relation seems to represent a shareable part-whole relation that makes no commitment on whether it is complete, or whether any or all parts are mandatory or optional.

However, considering that the interpretations of *Concern*, *Assessment*, *Goal*, *Requirements* and *Principles* all rely on propositions (*Concern*, *Assessment* and *Goal* are propositions and *Requirements* and *Principles* are desires which have as content propositions), this relationship can be defined more accurately as a logic relation between propositions. More specifically, the propositional content Pi of each of the n fine-grained model elements appears as a term in the propositional content P of the composed model element, formally, ($P \leftrightarrow P_1 \bullet_1 P_2 \bullet_2 \ldots \bullet_{n-1} P_n \bullet_n Z$), where $\bullet_i$ represents either the disjunction or the conjunction operator and an optional term Z represents any other proposition that may be used to derive P and is not explicitly modeled in the aggregation. Z captures the ambiguity regarding the notion of incompleteness that is associated with the *aggregation* concept.

The ability to model an incomplete *aggregation* is particularly useful, since one may not be able to list all possible decompositions of an intention, many of which may be unknown at the time of modeling. Furthermore, a modeler may choose to omit the least relevant decompositions when faced with an *aggregation* with numerous decompositions. Nevertheless, the language could also have opted to distinguish between an incomplete and a complete *aggregation*. In a complete *aggregation* ($P \leftrightarrow P_1 \bullet_1 P_2 \bullet_2 \ldots \bullet_{n-1} P_n$) for which the satisfaction of all P_i entails the satisfaction of P. The fact that the operators $\bullet_i$ are undefined is a characteristic of the language. This is different from certain goal modeling approaches (such as Tropos) (BRESCIANI, PERINI, *et al.*, 2004) where there is a distinction between the AND- and the OR-decomposition of goal. This is important and relevant on defining the organization's strategies to accomplish its goals, to illustrate, on the example given in Figure 30, the organizational goal 'Reduce workload employees' is to be accomplished by either reducing manual work and or by reducing interaction with customer? The stategy needed to accomplish one of the goals is different from the strategy to accomplish both goals.

The ME might have chosen to omit this language constructs for the sake of simplicity and ease of use, but from the perspective of expressiveness and clear semantics, the ME should be specialized with additional elements to capture the distinction between AND- and OR-decomposition.

The interpretation of the *aggregation* relationship for the *stakeholder* concept depends on the interpretation of the *stakeholder* concept, which, as discussed, can be represented either as an agent individual or as an agent universal. In the case in which it represents an agent individual, the aggregation represents either:

(i) a part-whole relationship between an institutional agent (the whole) and the aggregated agents (the parts) called componentOf (GUIZZARDI, 2005);

(ii) A part-whole relationship between a collective agent (the whole) and the aggregated agents (the parts) called memberOf (GUIZZARDI, 2005).

The distinction, however, cannot be expressed in the *stakeholder* concept. In the ME this is informally captured in the labels and description of a *stakeholder*. In the case in which the *stakeholder* concept represents an agent universal, the interpretation is of the *aggregation* relationship as a shareable part-whole relation that makes no commitment on whether it is complete, or whether any or all parts are mandatory or optional. When instantiated this relationship will either represents the componentOf relation or the memberOf relation depending on the natures of the related elements.

Now, the focus is to analyze the *specialization* relationship when applied to *Concern*, *Assessment*, *Goal*, *Requirements* and *Principles*. Considering that the interpretations of *Concern*, *Assessment* and *Goal* are propositions and *Requirements* and *Principles* are desires which have as content propositions, we can define this relationship as a logic relation between propositions. More specifically, the set of situations that satisfy the propositional content P' of the specialized model element is a subset of the set of situations that satisfy the propositional content P model element being specialized. Thus, formally, $P \rightarrow P'$.

When considering the *specialization* relationship applied to the *stakeholder* concept, it is understood that this relationship can only be applied meaningfully in the case in which a *stakeholder* is used to represent an agent universal. In that case, all instances of the specialized agent universal are instances of the agent universal being specialized. In the cases in which the *stakeholder* is an agent individual, the *specializes* relationship would allow some undefined relations, such as: (i) a physical individual to be specialized by another physical individual,e.g., 'John' specializes 'Maria' ; (ii) a physical individual to be specialized by an institutional agent,e.g., John being specialized by University of Twente; (iii) an institutional agent being specialized by an institutional agent (iii) University of Twente specializes Coca-Cola; and (iv) an institutional agent being specialized by a physical individual, e.g, 'Carlos' specializes 'University of Twente'.

5.5 SUMMARY OF THE ONTOLOGICAL INTERPRETATION

This section presents a summary of the ontological Interpretation. Table 3 presents the ME concept, a context information to each concept and its ontological interpretation and Table 4 presents key findings and possible solutions.

Table 3- Summary of the Ontological Interpretation

ME Concept	Context Information	Ontological Interpretation
Stakeholder	"Stakeholder can be an individual, team, organization or classes thereof."	Agent or universal that can be instantiated by agents.
Concern and Stakeholder has Concern relation	A *concern* is "some key interest that is important to certain stakeholders in a system". *Concerns* can be shared.	The propositional content of an agent's belief. The proposition in the case of a *stakeholder*'s *concern* refers to the importance that the *stakeholder* ascribes to certain *concern*.
Assessment and Assessment analyzes Concern relation	An *assessment* "represents the outcome of the analysis of some concern".	A belief of an undetermined agent. The propositional content of the belief refers specifically to the properties or characteristics believed to be important (the object of the *concern*) - in a specific situation
Goal	"A goal is some end that a *stakeholder* wants to achieve"	A goal of an undetermined agent. A goal is the propositional content of an agent's intention.
Goal addresses Assessment	"new or adapted business goals" can be defined to address Assessments	Some agent adopts a goal committing to pursue a desired situation. The *goal* is motivated by the wish to change the situation that is revealed by the *assessment*. This wish can be understood as an implicit desire that is not satisfied in the situation revealed by the

		assessment.
Requirement	"Represents a desired property that must be realized by some specific system." However, *requirements* must be satisfied if the system is to exist.	A desire whose propositional content is formulated as a normative description, stating that if a system is to exist, then it must satisfy the *requirement's* proposition.
Principle	"Represents a general desired property that applies to any system in a given context."	Desire. The propositional content *P* of the desire is the result of the application of the predicate *Q* on all systems in a given context, i.e., $P \equiv \forall s\, ((System(s) \wedge ContextPrinciple(s)) \rightarrow Q(s))$, where *System* holds for all systems, *ContextPrinciple* holds for all systems in the context of application of the principle and *Q* holds for the systems that exhibit the desired properties stated in the principle.
Contributes to relation	The realization of some intention A contributes positively or negatively to the realization of another intention B.	Less/more effort is required to reach a situation that satisfies the propositional content of *B* from the situation that satisfies the propositional content of *A*.
Conflicts to relation	Used to describe that two intentions cannot be realized both, and as such are in conflict with each other.	There are no events (or complex sequences thereof) that would lead to a situation in which the propositional content of both *intentions* (*Principle/Goal/Requirement*) would hold.
Core element realizes Requirement	The presence of this relationship indicates that the *stakeholder* believes that the core element exhibits the *required* properties.	The propositional content of the desire (corresponding to the *requirement R*) *is* satisfied in all the situations which include the stated *core element.*
Requirement realizes Goal	Whenever the *requirement* is satisfied the *Goal* is also satisfied.	The goal (proposition) *G* is satisfied in all situations that satisfy the propositional content *R* in the requirement.
Principle realizes	Whenever the *principle* is	The goal *G* is satisfied in all situations that satisfy

Goal	satisfied the *Goal* is also satisfied.	the propositional content *P* of the principle
Aggregation Relationship (between elements of the same type)	In case of *Concern, Assessment, Goal, Principle and Requirements* since they rely on propositions the aggregation relation is interpreted as a relation between propositions.	The propositional content P_i of each of the *n* fine-grained model elements appears as a term in the propositional content *P* of the composed model element, formally, $(P \leftrightarrow P_1 \bullet_1 P_2 \bullet_2 \ldots \bullet_{n-1} P_n \bullet_n Z)$), where $\bullet_i$ represents either the disjunction or the conjunction operator and an optional term *Z* represents any other proposition that may be used to derive *P* and is not explicitly modeled in the aggregation.
Aggregation Relationship (between *stakeholders*)	When *stakeholder* represents an agent	(i) A part-whole relationship between an institutional agent (the whole) and the aggregated agents (the parts) called componentOf, or (ii) a part-whole relationship between a collective agent (the whole) and the aggregated agents (the parts) called memberOf (GUIZZARDI, 2005)
	When *stakeholder* represents an agent universal	When instantiated this relationship will either represents the componentOf relation or the memberOf relation depending on the natures of the related elements.
Specialization Relationship (between *intentions* of the same type)	Considering that the interpretations of *Concern, Assessment, Goal, Requirements* and *Principles* rely on propositions, we can define this relationship as a logic relation between propositions.	The set of situations that satisfy the propositional content *P* of the specialized model element is a subset of the set of situations that satisfy the propositional content *P'* model element being specialized. Thus, formally, $P \rightarrow P'$.
Specialization Relationship (between *stakeholders*)	We believe that the *specialization* relationship can only be applied meaningfully to the *stakeholder* concept, whenever a *stakeholder* is used to represent an agent	All instances of the specializing agent universal are instances of the specialized agent universal.

	universal being specialized.	
Requirement specializes principle	"indicates that an object is a specialization of another object. In the current context, this relationship is used in particular to describe that a principle is specialized into a requirement".	A predicate *Q* (which contains the properties stated in the principle) is satisfied (when applied to that specific system) in all situations that satisfy the propositional content *R* in the requirement.

Table 4- Key Findings and Possible Solutions

ME Concept	Key Findings	Possible Solution
Stakeholder	Construct Overload. Different possible semantic interpretations to the same concept.	Specialize the stakeholder concept into at least two different elements, one to represent an agent and the other to represent a universal that can be instantiated by agents. (This solution could trigger other similar problems, since agents can be used to instantiate different sorts of universals, again suggesting new specializations of the stakeholder concept.)
Concern and Stakeholder has Concern relation	The concern itself, i.e., that which is important, is only represented in the ME as the label of the modeling concept. The properties or the characteristics which are supposed to be the focus of attention are not explicitly represented in the model.	Since the properties or characteristics which are supposed to be the focus of the concern's attention are not explicitly represented in the language, at this moment a detailed interpretation of the possible contents of the proposition associated with a stakeholder's concern is not possible within the scope of this thesis.
Assessment and Assessment analyzes Concern relation	The ME does not indicate who performs the assessment. Thus, while the propositional content of the belief is settled (informally in the assessment's label), the identity of the agent remains	The ME might lack a construct to relate assessments and stakeholders - to determine the identity of the agent that holds the belief. A possible solution is the addition of a construct to relate assessments and

	undetermined.	stakeholders.
Goal	The ME does not indicate the stakeholder who has the goal. Thus, while the propositional content of the intention is settled (informally in the goal's label), the identity of the agent remains undetermined	The ME might lack a construct to relate goals and stakeholders, to determine the identity of the agent that holds the intention of bringing about the goal. A possible solution is the addition of a construct to relate goals and stakeholders.
Principle	The ME does not relate principles and stakeholders, thus it is understood that the agent that is implied in a principle is the enterprise as a whole. This may be relevant when conflicting principles are considered for different stakeholders in an intra-enterprise or inter-enterprise setting	The ME might lack a construct to relate principles and stakeholders, to determine the identity of the agent that holds the desire. A possible solution is the addition of a construct to relate principles and stakeholders.
Aggregation Relationship (between elements of the same type)	In case of *Concern, Assessment, Goal, Principle and Requirements.* i-The ME does not distinguish between an incomplete and a complete aggregation. ii-The language does not distinguish between the AND- and the OR-decompositions.	The ME might have chosen to omit these language constructs for the sake of simplicity and ease of use, but from the perspective of expressiveness and clear semantics, the ME should be specialized with additional elements to capture these distinctions.
Aggregation Relationship (between *stakeholders*)	There are two possible interpretations whenever a *stakeholder* represents an agent: (i) A part-whole relationship between an institutional agent (the whole) and the aggregated agents (the parts) called componentOf, or (ii) a part-whole relationship between a collective agent (the whole) and the aggregated agents (the parts) called memberOf (GUIZZARDI, 2005).	The ME might have chosen to omit language constructs for the sake of simplicity and ease of use, but from the sole perspective of expressiveness and clear semantics, the ME should be specialized with additional elements to capture these distinctions.
	When *stakeholder* represents an agent universal the interpretation is of the aggregation relationship as a shareable	The ME might have chosen to omit these distinctions in the aggregation relationship for the sake of simplicity and ease of use, but

	part-whole relation that makes no commitment on whether it is complete, or whether any or all parts are mandatory or optional. When instantiated this relationship will either represents the componentOf relation or the memberOf relation depending on the natures of the related elements.	from the perspective of expressiveness and clear semantics, the ME should be specialized with additional elements to capture these distinctions.
Specialization Relationship (between *stakeholders*)	In the cases in which the stakeholder is an agent individual, the specializes relationship would allow some undefined relations, such as: (i) a physical individual to be specialized by another physical individual; (ii) a physical individual to be specialized by an institutional agent (iii) an institutional agent being specialized by an institutional agent; and (iv) an institutional agent being specialized by a physical individual.	The ME should restrict its syntax to avoid the use of relations with no meaningful use. The examples should in fact be considered as syntactically incorrect models.

5.6 EXAMPLE IN TERMS OF UFO

This subsection applies the proposed interpretation to the example in Figure 21 (shown on Figure 30 for convenience).

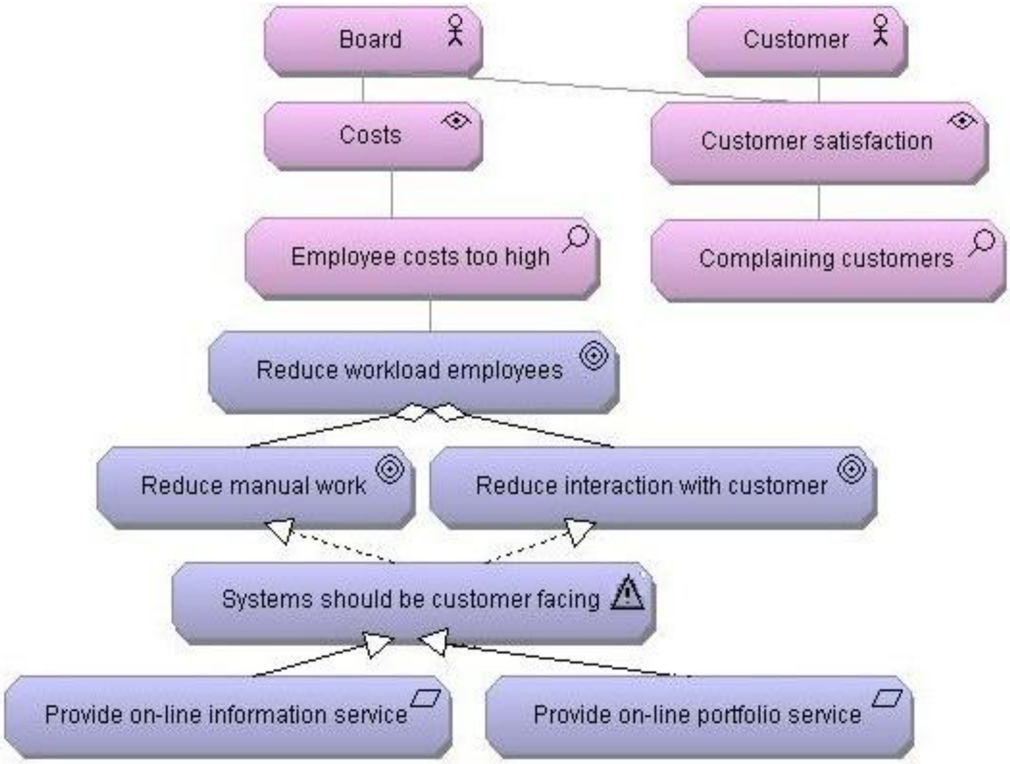


Figure 30- ME Example

The example shows the 'Board' stakeholder, which we interpret here as representing an Institutional Agent (the organization's board) and the 'Customer' stakeholder, which we interpret here as representing either an Agent Universal or Social Role which will be instantiated by Agents that are customers of the organization.

The model shows that the Institutional agent 'Board' has a belief whose content refers to the importance of 'Costs' and a belief whose content refers to the importance of 'Customer Satisfaction'. The model shows that every agent instantiating 'Customer' holds a belief with the same propositional content (on the importance of 'Customer Satisfaction').

The model also shows that some assessment has been conducted considering the 'Customer Satisfaction' concern, leading to the conclusion that there are 'Complaining Customers'. In other words, some (undetermined) agent holds a belief whose propositional content refers specifically to properties or characteristics believed to be important ('Customer Satisfaction'). In this case, it is unclear as to whether the content of the assessment is shared by the identified stakeholders, so we cannot know whether the belief is held by each agent instantiating 'Customer', by the 'Board' or by all stakeholders. We reaffirm that the ME is not expressive enough to satisfactorily represent the cases in which different and possibly contradictory assessments are made by different stakeholders. These cases may be frequent which would justify adding language constructs to relate stakeholders and assessments. Additionally, in light of the proposed interpretation, careful analysis of the labels of the assessment modeling element reveals a rather indirect choice of text in the example ('Complaining Customers'), with no explicit mention of the concern ('Customer Satisfaction'). The interpretation we have provided suggests that a more explicit reference to the concern would be desirable (e.g., renaming the assessment to 'Customer Satisfaction is low').

The goal 'Reduce workload employees' can be interpreted as representing that some (undetermined) agent holds the goal of reducing the workload of employees and commits on acting to pursue a situation that satisfies the propositional content of his/her intention. The goal 'Reduce workload employees' aggregates the goals 'Reduce manual work' and 'Reduce interaction with customer'. According to our interpretation, the model suggests the existence of a logical relation between the content of 'Reduce workload employees' and the contents of 'Reduce manual work' and 'Reduce interaction with customer'. As discussed earlier, the model does not precisely identify this relation, thus it is possible that the modelers mean that both sub goals need to be attained for the aggregated goal to be satisfied, or that one of the sub goals needs to be attained for the goal to be satisfied.

The model further states that principle 'Systems should be Customer Facing' realizes the goals 'Reduce manual work' and 'Reduce interaction with customer'. This means that whenever the desired proposition 'Systems should be Customer Facing' is satisfied the goals are satisfied. (And, thus, 'Reduce workload employees' is satisfied.)

The relationship of specialization between the principle 'Systems should be customer facing' and the requirements 'Provide on-line information service' and 'Provide on-line portfolio service' states that whenever a system meets the requirements proposition, this system complies to the principle proposition, i.e., the principle proposition holds when applied to that specific system. This model shows that if a system provides on-line information service or provides on-line portfolio service it is a customer facing system. In the scope of the model, if the enterprise's systems satisfy the requirements, then the overarching goal 'Reduce workload employees' is satisfied.

5.7 RELATED WORK

An early approach to have suggested the use of foundational ontologies in determining the semantics of modeling languages was discussed by Evermann and Wand in (EVERMANN and WAND, 2001). The authors propose a mapping of UML constructs to the Bunge-Wand-Weber (BWW) ontology and show that by mapping UML constructs to well-defined ontological concepts, clearer semantics can be obtained. In (GUIZZARDI, 2005), the UML class diagrams have been subject to thorough review using UFO as a semantic foundation. The author has created the so-called OntoUML as a result of this review; an ontologically well-founded conceptual modeling language.

A number of enterprise modeling approaches have been subject to ontology-based analysis in recent years. With respect to enterprise business processes (RECKER, INDULSKA, *et al.*, 2010) did an ontological analysis to uncover ontological deficiencies regarding a business process modeling language, its user's experiences and easy of use when employed in organizations. The authors' found nine ontological deficiencies related to modeling when using the Business Process Modeling Notation (BPMN). Another approach to apply ontological analysis on business process modeling languages is that of Zur Muehlen and Indulska (2010) who used the BWW ontology to compare the semantic representational capabilities of some process and rule modeling languages. The authors focus on the aspects of maximum ontological completeness and minimum ontological overlap, mapping the elements of the business rules specifications against the constructs of the BWW ontology. The outcome of this study shows constructs overloads, constructs redundancy and constructs deficits in the studied languages[13]. The study shows that none of the studied languages is internally complete with respect to the BWW representation model. Based on the research results, the authors speculate that business rule modeling languages might be used in conjunction with business process modeling languages to better represent enterprise operations. Differently from the approach employed in this thesis, they focus solely on a mapping of the language to the foundational ontology and retrain their conclusions on that. Our approach, on the other hand, focuses on the the language itself, addressing its semantics. Our approach also maps the language to a foundational ontology to uncover ontological deficiencies of the language.

With respect to more comprehensive enterprise architecture approaches, in (SANTOS JR., ALMEIDA, *et al.*, 2010) and (SANTOS JR., ALMEIDA and GUIZZARDI, 2010), the authors performed an ontological analysis and interpretation of the ARIS framework. The authors' intention was to define the semantics of ARIS concepts and relationships in terms of UFO. Some problems regarding the ARIS Method were exposed and possible solutions to these problems were proposed. In (ALMEIDA, GUIZZARDI and SANTOS JR, 2009), the authors have used UFO as a semantic foundation to analyze several enterprise modeling approaches with respect to the representation of role-related concepts (what is called "active structure" in the ArchiMate framework, or that which fills the "who" column of Zachman's framework). Several cases of semantic overload and lack of expressiveness have been detected.

Concerning specifically analyses on the goal domain, Guizzardi et al. (2007) have analyzed the Tropos language and proposed an interpretation of the language concepts in terms of UFO. Still on the goal domain, Cardoso et al. (2010) have discussed the semantic interpretation of a subset of Tropos and the goal modeling constructs in the ARIS framework. The authors have semantically integrated Tropos and ARIS defining the semantics of both languages in terms of the UFO and have provided a correspondence between them. Also, Franch et al. (2011) use UFO to analyze the ontological underpinnings of the means-end-link, an i* core element. The work has been proposed to serve as a basis for a community

---

[13] The paper explicitly shows constructs lack and overlap analysis - mapped to construct deficit and construct redundancy. Construct overload are not explicitly shown in the paper, but can be extracted analyzing the research results.

agreement about what a means-end link exactly is, since i* definition is "neither fully detailed nor formal, and researchers may have interpreted the same constructs in different ways" (FRANCH, GUIZZARDI, *et al.*, 2011). The research long-term idea is to use the foundational ontology as a reference model to which i* metamodel concepts should be mapped to.

With respect to the completeness and correctness of the ontological foundation used, the ontological foundations mainly used on ontological analyses are the BWW (EVERMANN and WAND, 2001b), (WAND and WEBER, 1990), (WEBER, 1997) ontology and the UFO ontology. In (GUIZZARDI, 2005) and (GUIZZARDI, HERRE and WAGNER, 2002) the authors discusses the BWW ontology and the UFO ontology. The authors criticize the BWW ontology on respect of its expressiveness. In (GUIZZARDI, 2005), the author compares UFO with BWW and concludes that UFO has "a much richer system of ontological distinctions". Further, the UFO ontology has social concepts, which are essential for the work developed in this thesis.

CHAPTER 6. CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

Enterprise Architecture modeling approaches structure the enterprise in terms of various related architectural domains or viewpoints (IEEE COMPUTER SOCIETY, 2000), (DIJKMAN, 2006) each of them focusing on specific aspects of the enterprise. The motivation domain is concerned with "why" an Enterprise Architecture is defined the way it is. This domain addresses enterprise objectives in a broad scope ranging from high-level statements expressing the goals of an enterprise to declarations of requirements on business processes, services and systems (CARDOSO, ALMEIDA and GUIZZARDI, 2010) and impacts the whole organization.

Although the motivation domain importance for Enterprise Architecture has been recognized since at least two decades (SOWA and ZACHMAN, 1992), few comprehensive enterprise modeling approaches addresses the motivation domain (CARDOSO, SANTOS JR., *et al.*, 2010). The incorporation of this domain in a comprehensive Enterprise Architecture modeling language is being developed with the upcoming Motivational Extension to ArchiMate (QUARTEL, ENGELSMAN, *et al.*, 2009). The Motivation Extension includes concepts such as concerns, assessments, goals, principles and requirements in ArchiMate.

This thesis defends that careful definition of the semantics of the Motivation Extension to ArchiMate is indispensable, especially since the motivation domain concepts addresses subjective aspects of the enterprise, which have eluded clear semantics definition in the past. (CARDOSO, ALMEIDA and GUIZZARDI, 2010).

The ontological analysis performed on this work has revealed some issues with the Motivation Extension. With respect to lack of expressiveness the ontological analysis allowed the identification of absence of constructs in the language to relate the *stakeholder* concept to the concepts of *assessment*, *principle*, *requirement* and *goal*. As a result, the ME is currently unable to capture the cases in which different *assessments* are performed by different *stakeholders* with shared *concerns*. Analogous problems occur when dealing with *stakeholders* and *principles*, *requirements* or *goals*. This is problematic as many of the claimed benefits of goal modeling stem from the possibility of contrasting the goals of the various enterprise *stakeholders*. This requires at least the identification of the relation between *stakeholders* and their *goals*. The importance of the identification of the *stakeholders* may be even more evident in an inter-organizational model, since different organizations will likely have different goals, as well as will likely apply different *principles* and *requirements* on their *systems*. To solve this problem, the language should relate the stakeholder to the other motivational elements.

The ontological analysis also showed that the language has a construct overload problem and lacks the expressiveness to distinguish between a *stakeholder* representing a specific agent and a *stakeholder* representing any agent instantiating a particular agent universal.

Through ontological analysis we have also been able to identify some problems resulting from the usage of the generic relations of ArchiMate in the Motivation Extension, more specifically, the consequences of the unqualified usage of the *specialization* relation between *stakeholders*. Since the *stakeholder* construct is used to represent entities of different ontological natures (agent individuals and agent universals), this relation does not always have a meaningful interpretation. This problem in the Motivation Extension becomes evident when a *stakeholder* representing an agent universal (e.g., Professor) specializes a *stakeholder* representing an agent individual (e.g., John); or when a *stakeholder* representing an agent individual (e.g., Mary) specializes another *stakeholder* also representing an agent individual (e.g., John).

Another issue raised by the ontological analysis is the lack of sophistication in *goal* refinement relations. The extension is currently unable to distinguish between AND- and OR-*goal* decompositions (nor AND or OR decompositions for principles, assessments, requirements and concerns). This distinction is instrumental to *goal* analysis, (e.g., one goal decomposed into three specialized goals should mean that the first is accomplished whenever the latters are accomplished - both, combined or just one of the latters is needed to the first be accomplished?) and is one of the basic distinctions discussed in the goal modeling literature (see, e.g., (BRESCIANI, PERINI, *et al.*, 2004)).

6.2 LIMITATIONS

This section discusses the limitations of the work described in this thesis.

In line with (SANTOS JR, ALMEIDA and GUIZZARDI, 2012, submitted), we should note that the main role of the ontological analysis has been to provide us with a rigorous framework to analyze the Motivation Extension proposed to ArchiMate. In this sense, the ontological analysis should be seen as a tool for hypothesis formulation, and the recommendations that were identified here should be considered as hypothesis subject to further examination. In particular, one should consider the pragmatic impact of amendments on the language and its users. It has not been possible to evaluate the pragmatics on real users of the language, due to the fact that the motivation extension is still a proposal and there are no real users or practitioners of the extension. This fact impacts directly in the validation of the semantics described on the thesis. Also, the limited number of examples of usage might have an impact on the definition of the semantics presented here.

There must be also made clear that it was hard to uncover the original intended semantics for some of the concepts of the Motivation Extension. The white paper of the submitted proposal was not completely clear on this matter for each of the concepts and the designers' intentions were not always visible or easy to understand. In this thesis this risk has been mitigated since the work presented here had the collaboration of one of the proponents of the extension[14].

Further, it is not our aim to suggest that the terminology used in this thesis should replace the terminology currently used in the language, and there is no intention to imply that the UFO conceptualization should be exposed directly to language users.

Also, the work presented in this thesis is based on the UFO ontology and thus, the analysis conducted here is limited to the concepts described in the ontology. If the ontology is incomplete with respect to some phenomena of interest, the work described on this thesis would also have this limitation. In that case, the work presented here would not be able to fully identify the phenomena or its relations.

---

[14] The collaboration of the extension proponent Dick Quartel can be verified on the paper (AZEVEDO, ALMEIDA, *et al.*, 2011), in which he is one of the authors, and that presents part of the results of this thesis.

A problem that we have dealt with was on the concept of effort, which has no direct mapping on UFO. This concept has been used in order to define the semantics of the contribution relationship. Another possible limitation of UFO is the lack of direct relation and relationships on the concept of a particular belief on a particular desire and on the relation of a particular desire on a particular intention. These social concepts appear to have a relation to each other, as in (FERRARIO and OLTRAMARI, 2005) and in (DIGNUM, DIGNUM and JONKER, 2009), in the sense that an agent would firstly need to have a belief about something, to them have a desire with a subject related to that particular belief, to finally have an intention on doing something to achieve the situation stated on that particular desire. Neither the relation between the original belief that led to a desire nor the relation between a particular desire and the intention to satisfy the desire can be expressed in UFO.

6.3 FUTURE WORK

Considering what has been presented on this thesis and the results of the ontological analysis of the Motivation Extension, the following lines of research have been envisioned as promising directions.

i. A thorough ontological analysis of ArchiMate's core concepts.

This ontological analysis would be able to define precise semantics for the ArchiMate language. The analysis might reveal some semantic issues with the language. The analysis focus must also consider the implications of the semantics of the core ArchiMate concepts to the Motivation Extension. This would be a step towards the integration of the enterprise as a whole. Some other works that could be then realized follow the directions on the semantic analysis of completion and traceability on the enterprise.

ii. An ontological interpretation for the TOGAF content metamodel (THE OPEN GROUP, 2009)

This interpretation aims at systematically addressing the alignment of ArchiMate and TOGAF. This work would enable the semantic integration and alignment of TOGAF and

ArchiMate. When used together, these two approaches would enable the idea to have coherent descriptions of the enterprise towards its semantic integration.

iii. An empirical study with practitioners on the semantics of the Motivation Extension.

The empirical study would be able to analyze the semantics given to the language by the users when they are working with the extension both with access on the semantics defined on this thesis and only with the specification. This study could then be collapsed to the clarity of the semantics defined here. The study would also verify the users perception of the ontological deficiencies reported on this thesis.

iv. An investigation on a definition of a well-founded dialect for organizational structure modeling.

Further investigation could be carried in order to lead to the definition of a well-founded dialect for organizational structure modeling, which itself would be an object of empirical validation.

v. An ontological analysis and interpretation of BPMN (OBJECT MANAGEMENT GROUP, 2010).

This analysis has, firstly, the objective to precisely define semantics to BPMN. Another objective would be defining how to use BPMN in tandem with ArchiMate and the Motivation Extension. This would allow one to model detailed business process views while relating those to the overall enterprise goals. That approach would be able to give semantics on traceability.

vi. An analysis on the real-world social concepts to account for additional social concepts on the Unified Foundational Ontology.

A study on the social concepts of the real-world and its implication on UFO should be considered. A full analysis on the UFO social concepts and its relations should be carried in order to study concepts such as effort and also the relation of beliefs, desires and intentions to each other and to commitments and effort. This work would problably extend UFO and enable it to comprise other aspects of the social domain.

This thesis has been proposed in the context of the Ontology and Conceptual Modeling Research Group (NEMO) and in such is part of a central idea to have coherent descriptions of the enterprise towards its semantic integration.

CHAPTER 7. REFERENCES

ALMEIDA, J. P. A.; GUIZZARDI, G.; SANTOS JR, P. S. Applying and Extending a Semantic Foundation for Role-Related Concepts in Enterprise Modelling. **Enterprise Information Systems**, 3, 2009. 253-277.

ANDERSSON, B. et al. Enterprise Sustainability through the Alignment of Goal Models and Business Models. **Proceeding of 3rd International Workshop on Business /IT-Alignment and Interoperability (BUSITAL'08)**, 2008.

ARANGO, G. **Domain Analysis Methods**. Workshop on Software Architecture. Los Angeles, USA: USC Center for Software Engineering. 1994.

ARANGO, G.; WILLIAMS, G.; ISCOE, N. **Domain Modeling for Software Engineering**. Proceedings of ICSE 1991: The International Conference on Software Engineering. Austin, Texas: ACM Press. 1991.

AZEVEDO, C. L. B. et al. An Ontology-Based Semantics for the Motivation Extension to ArchiMate. **Proceedings of the 15th IEE International EDOC Conference (EDOC)**, 2011.

BRESCIANI, P. et al. Tropos: An Agent-Oriented Software Development Methodology. **Autonomous Agents and Multi-Agent Systems**, v. 8, n. 3, p. 203-236, May 2004.

CAMPBELL, K. **Abstract Particulars**. Blackwell: Cambrigde, 1990.

CARDOSO, E. C. S. **On the Alignement Between Goal Models and Enterprise Models with an Ontological Account (Masther Thesis)**. Vitória: [s.n.], 2010.

CARDOSO, E. C. S. et al. Semantic Integration of Goal and Business Process Modelling. **IFIP: International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS 2010)**, 2010.

CARDOSO, E. C. S.; ALMEIDA, J. P. A.; GUIZZARDI, R. S. S. On the Support for the Goal Domain in Enterprise Modelling Approaches. **Enterprise Distributed Object Computing Conference Workshops (EDOCW)**, Oct 2010. 335-344.

CLANCY, W. J. The Knowledge Level Reinterpreted: Modelling Socio-Technical Systems. **International Journal of Intelligent Systems**, v. 8, p. 33-49, 1993.

DEGEN, W. et al. **GOL:** toward an axiomatized upper-level ontology. Proceedings of the international conference on Formal Ontology in Information Systems - FOIS '01. Ogunquit, Maine, USA. : [s.n.]. 2001. p. 34-46.

DIGNUM, F.; DIGNUM, V.; JONKER, C. M. Towards Agents for Policy Making. **Lecture Notes in Computer Science**, v. 5269/2009, p. 141-153, 2009. ISSN 10.1007/978-3-642-01991-3_11.

DIJKMAN, R. **Consistency in Multi-Viewpoint Architectural Design.** [S.l.]: University of Twente, 2006.

DODAF. **DODAF:** DoD Architecture Framework Version 1.5. Product Descriptions. ed. [S.l.]: [s.n.], v. II, 2007.

EERTINK, H. et al. **A Business Process Design Language**. Proceedings of the 1st World Congress on Formal Methods. Toulouse, France: [s.n.]. Sept. 1999.

EVERMANN, J.; WAND, Y. Towards Ontologically Based Semantics for UML Constructs. **Proceedings of the 20th International Conference on Conceptual Modeling**, Nov 2001. 27-30.

EVERMANN, J.; WAND, Y. **An ontological examination of object interaction in conceptual modeling**. Proceedings of the Workshop on Information Technologies and Systems WITS'01. New Orleans: [s.n.]. 2001b.

FERRARIO, R.; OLTRAMARI, A. **Towards a Computational Ontology of Mind**. Proceedings of the 3rd International Conference on Formal Ontology in Information Systems (FOIS). Torino: [s.n.]. 2005. p. 1-9.

FRANCH, X. et al. **Ontological Analysis of Means-End-Links**. CEUR Proceedings of the 5th International i* Workshop (iStar 2011). Trento: [s.n.]. 2011. p. 37-42.

GUARINO, N. Formal Ontology and Information Systems. **Proceedings of the International Conference on Formal Ontology and Information Systems**, Trento, Italy, June 1998. 3-15.

GUIZZARDI, G. **Ontological Foundations for Structural Conceptual Models, Ph.D. Thesis**. University of Twente: CTIT PhD-thesis, 2005.

GUIZZARDI, G.; FALBO, R.; GUIZZARDI, R. S. S. Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. **Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environment**, Recife, 2008.

GUIZZARDI, G.; HERRE, H.; WAGNER, G. **On the General Ontological Foundations of Conceptual Modeling**. Proc. of 21st Intl. Conf. on Conceptual Modeling (ER 2002). Berlin: Springer-Verlag. 2002.

GUIZZARDI, G.; WAGNER, G. **Some Applications of a Unified Foundational Ontology in Business Modeling**. M. Rosemann P. Green (Orgs.). [S.l.]: IDEA Publisher. 2005. p. 345-367.

GUIZZARDI, R. S. S. et al. Towards an Ontological Account of Agent-Oriented Goals. **Software Engineering for Multi-Agent Systems V**, 4408, 2007. 148-164.

GUIZZARDI, R. S. S.; GUIZZARDI, G. Ontology-Based Transformation Framework from Tropos to AORML. In: GIORGINI, P., et al. **Social Modeling for Requirements Engineering, Cooperative Information Systems Series**. Boston: MIT Press, 2011. p. 547-570.

IEEE COMPUTER SOCIETY. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. **IEEE Std 1471-2000**, New York, 2000.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO) / INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC). **Systems and software engineering -- Recommended practice for architectural description of software-intensive systems**. [S.l.]: [s.n.], v. ISO/IEC 42010:2007, 2007.

ISO - INTERNATIONAL ORGANIZATION FOR STANDARD. **ISO/IEC 19793:** 2008, Information technology — Open distributed processing — Use of UML for ODP. [S.l.]: [s.n.], 2008.

JARDINE, D. A. **The ANSI/SPARC DBMS model**. proceedings of the second SHARE Working Conference on Data Base Management Systems. Montreal, Canada: [s.n.]. 1976.

JONKERS, H. et al. Concepts for Modelling Enterprise Architectures. **International Journal of Cooperative Information Systems**, 3, 2004. 257–287.

JONKERS, H. et al. Enterprise architecture: Management Tool and Blueprint for the Organization. **Information Systems Frontiers**, The Netherlands, v. 8, n. Springer-Verlag, p. 63-66, 2006.

LANKHORST, M. et al. **Enterprise Architecture at Work - Modelling, Communication, and Analysis**. [S.l.]: Springer-Verlag, 2005.

LOWE, E. J. **The possibility of Metaphysics:** Substance, Identity and Time. [S.l.]: Oxford University Press, 2001.

MEALY, G. H. **Another Look at Data**. Proceedings of the November Fall Joint Computer Conference. New York, NY, USA: ACM. 1967. p. 14-16.

MOODY, D. The 'Physics' of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. **IEEE Transactions on Software Engineering**, v. 35, p. 756-779, 2009.

OBJECT MANAGEMENT GROUP. **Business Process Modelling Notation BPMN 2.0**. [S.l.]: [s.n.], 2010.

QUARTEL, D. A. C. et al. A Goal-Oriented Requirements Modelling Language for Enterprise Architecture. **Proceedings of 13th IEEE International EDOC Conference (EDOC 2009)**, 2009.

QUARTEL, D.; ENGELSMAN, W.; JONKERS, H. **Whitepaper:** Archimate Extension for Modeling and Managing Motivation, Principles and Requirements in TOGAF, W109. [S.l.]: The Open Group, 2011.

QUARTEL, D.; ENGELSMAN, W.; JONKERS, H. **Whitepaper:** Archimate Extension for Modeling and Managing Motivation, Principles and Requirements in TOGAF, W109. [S.l.]: The Open Group, Feb 2011.

RECKER, J. et al. The ontological deficiencies of process modeling in practice. **European Journal of Information Systems**, 19, June 2010. 501-525.

RECKER, J. et al. Do Ontological Deficiencies in Modeling Grammars Matter? **MIS Quarterly**, MN, USA, v. 35(1), p. 57-79, March 2011.

SANTOS JR, P. S.; ALMEIDA, J. P. A.; GUIZZARDI, G. An Ontology-Based Analysis and Semantics for Organizational Structure Modeling in the ARIS Method. **Information System**, 2012, submitted.

SANTOS JR., P. S. et al. **An Ontology-Based Semantic Foundation for ARIS EPCs**. Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10). Sierre: [s.n.]. 2010. p. 124-130.

SANTOS JR., P. S.; ALMEIDA, J. P. A.; GUIZZARDI, G. **An Ontology-Based Semantic Foundation for Organizational Structure Modeling in the ARIS Method**. Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International. Vitoria: [s.n.]. Oct 2010. p. 272-282.

SCHEER, A.-W. **ARIS – Business Process Modeling**. [S.l.]: Springer, 2000.

SCHNEIDER, L. **Formalised Elementary Formal Ontology**. LOA. Italy. 2002. (ISIB-CNR Technical Report 03/2002). online: http://www.loa-cnr.it/Publications.html.

SCHNEIDER, L. Designing Foundational Ontologies: The Object-Centered High Level Reference Ontology OCHRE as a Case Study. **22th International Conference on Conceptual Modeling (ER)**, 2003.

SMITH, B.; WELTY, C. **Ontology:** Towards a new synthesis. Formal Ontology in Information Systems. Ongunquit, Maine: ACM Press. 2001. p. iii-x.

SOWA, F.; ZACHMAN, J. Extending and formalizing the framework for information systems architecture. **IBM Systems Journal**, 31, 1992. 590-616.

TANG, A.; HAN, J.; CHEN, P. **A comparative analysis of architecture frameworks**. Software Engineering Conference, 2004. 11th Asia-Pacific. [S.l.]: [s.n.]. 2004. p. 640 - 647.

THE OPEN GROUP. **ArchiMate 1.0 Specification, C091**. [S.l.]: [s.n.], 2009.

THE OPEN GROUP. **ArchiMate(R) Version 1.0 Technical Standard**. [S.l.]: [s.n.], 2009.

THE OPEN GROUP. **The Open Group Architecture Framework (TOGAF) version 9.1**. [S.l.]: [s.n.], dec 2011.

THE OPEN GROUP. **The Open Group Architecture Framework (TOGAF) version 9**. [S.l.]: [s.n.], feb 2009.

THE UNITED STATES DEPARTMENT OF DEFENSE. **DoD Architecture Framework version 1.5 Volume I:** Definitions and Guidelines. [S.l.]: [s.n.], v. I.

VAN LAMSWEERDE, A.; LETIER, E. From Object Orientation to Goal Orientation: A Paradigm Shift for Requirements Engineering. **Proceedings of Radical Innovations of Software and Systems Engineering**, v. 2941/2004, p. 153-166, 2004.

VERMEER, M. W. W. **Semantic interoperability for legacy databases**. PhD Thesis. ed. The Netherlands: [s.n.], 1997.

WAND, Y.; WEBER, R. An ontological evaluation of systems analysis and design methods. **Information System Concepts: An Indepth Analysis**, North-Holland, 1989.

WAND, Y.; WEBER, R. **Mario Bunge's Ontology as a formal foundation for information systems concepts**. Atlanta: Weingartner, P. & Dorn, G. J. W, 1990.

WEBER, R. **Ontological Foundations of Information Systems**. Melbourne, Australia: Coopers & Lybrand, 1997.

WELTY, C.; GUARINO, N. Supporting ontological analysis of taxonomic relationships. **Data Knowl. Eng.**, v. 39, p. 51-74, 2001. ISSN 1.

ZACHMAN, J. A Framework for Information Systems Architecture. **IBM Systems Journal**, 1987. 276-292.

ZACHMAN, J. Enterprise Architecture: The Issue of the Century. **Database Programming and Design**, 1997. 44-53.

ZUR MUEHLEN, M.; INDULSKA, M. Modeling languages for business processes and business rules: A representational analysis. **Information Systems**, n. 35, p. 379-390, 2010.