Federal University of Espírito Santo

Roberto Carraretto

Separating Ontological and Informational Concerns: A Model-driven Approach for Conceptual Modeling

Vitória, Brazil

November, 2012

Roberto Carraretto

Separating Ontological and Informational Concerns: A Model-driven Approach for Conceptual Modeling

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo para obtenção do título de Mestre em Informática.

Orientador: Prof. Dr. João Paulo Andrade Almeida

Vitória, Brazil

November, 2012

ACKNOWLEDGMENTS

I am thankful to the following people:

- My mother, for the unconditional love and support;
- João Paulo, for the mentorship, for the challenging and intriguing dissertation topic and for being open to new ideas. I admire your intelligence, your character and your dedication;
- Izon, for the partnership, especially during the time-consuming courses that were required by the master's degree program;
- Giancarlo Guizzardi, for providing comments on early drafts of this work (as it contributed to correct its direction), for being part of the examination committee and for providing further comments to the final version of this thesis;
- Maria Luiza Machado Campos for being part of the examination committee and for providing comments to the final version of this thesis; and,
- My sister, my father and friends (especially Ricardo Julião).

“*The map is not the territory*”

ABSTRACT

Many authors have discussed the importance of ontological concerns in the development of information systems, emphasizing the benefits of ontology-based approaches to conceptual modeling tasks. A principled ontology-driven approach typically relies on the definition of a domain ontology and its use in subsequent phases of information system design and integration. Many of the challenges in the application of such an approach are related to addressing ontological concerns (defining the nature of phenomena of interest) and addressing informational concerns (defining the information demand about the phenomena of interest). In this thesis, we argue that ontological concerns should be clearly separated from informational concerns. We have observed that ontology-based approaches and information modeling approaches have been treated mostly in isolation, with the consequence that the relation between a domain ontology and an information model is still in need of clarification, despite the efforts of the formal ontology and the information modeling communities.

Therefore, in this thesis, we analyze conceptual modeling in terms of two levels, namely, the so-called ontological level and, what we call, the information level. Our initial effort is to characterize the information level in harmony with existing works on information modeling and to align our information level approach with the existing ones concerning the ontological level. Then, we strive to provide a model-driven approach in which a domain ontology addressing ontological concerns (at the ontological level) is used as a starting point for the definition of an information model addressing informational concerns (at the information level). Our model-driven approach is guided by several systematic informational decisions that we identify here, which assist the addressing of an information demand. We adopt a philosophically well-founded profile of the Unified Modeling Language (UML) class diagrams, called OntoUML, to represent domain ontologies. Further, we adopt UML class diagrams to represent information models in an object-oriented approach. Finally, we provide tool support for the model transformation from OntoUML to UML in order to operationalize the approach and show its technical feasibility.

CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

Information and Communication Technologies (ICTs), in their most primitive form, were mainly *recording* systems, consisting of writing and manuscript production. After the invention of printing, ICTs also became *communication* systems. Then, after the diffusion of computers, ICTs became likewise *processing* and *producing* systems. Thanks to this evolution, nowadays the most advanced societies highly depend on information-based assets, information-intensive services (especially business and property services, communications, finance and insurance, and entertainment), and information-oriented public sectors (especially education, public administration, and health care). (Floridi, 2010)

The task of computer scientists is to develop theories, tools and techniques for managing this information and making it useful. To use information, one needs to represent it, capturing its *meaning* and inherent *structure*. Such representations are important for communicating information between people, but also for building information systems that manage and exploit information in the performance of useful tasks (Mylopoulos, 1998). A particular important activity to arrive at such representations is *conceptual modeling*, which is defined as "the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication. Moreover, it supports structuring and inferential facilities that are psychologically grounded. After all, the descriptions that arise from conceptual modeling activities are intended to be used by humans, not machines" (Mylopoulos, 1992).

Nevertheless, conceptual modeling does not have a single facet. Some approaches claim to be focused on *structure* while some claim to be focused on *meaning*. On one hand, *information modeling* claims to be focused on structure, more specifically, the structure of information about phenomena of interest. Information modeling is mainly driven towards the development of information systems and databases. For instance, according to (Halpin & Morgan, 2008), their book is written "primarily for data modelers and database practitioners" but also for "anyone wishing to formulate the information structure of business domains in a way that can be readily understood by humans yet easily implemented on computers". The interest in structure is also manifested in the ultimate specification of information modeling, the information model (or conceptual schema), which "describes the structure or grammar of the business domain (e.g., what types of object populate it, what roles these play, and what constraints apply)" (Halpin & Morgan, 2008). On the other hand, *ontology-based conceptual modeling* claims to be focused on meaning. Ontology-based approaches follow the lines of the ontological level proposed by Guarino, i.e., "on the basis of formal

ontology, intended as a theory of *a priori distinctions*: (i) among entities of the world (physical objects, events, processes, ...), (ii) among the meta-level categories used to model the world (concepts, properties, states, roles, attributes, various kinds of part-of relations...)" (Guarino, 1994).

Throughout this thesis, we advocate that this claimed distinction between "meaning" and "structure" is not always made clear by conceptual modeling approaches. Nonetheless, we still believe that both facets of conceptual modeling have their importance and, once the distinction between them has been clarified, it is possible to relate both. In this work, we view conceptual modeling in terms of two levels, namely, the *ontological level* and the *information level*.

On one hand, the ontological level aims at building an agreement on metaphysical aspects of a domain, capturing the categories of being that are assumed to exist in that domain. The ontological level supports the understanding about a domain in a way that is largely independent of structures that underlie exchanged information about that domain. On the other hand, the information level addresses unavoidable aspects of information manipulation in the scope of a domain. This effort assists (but is not limited to) information systems development. At the information level, there are several *informational concerns* that must be addressed. All those concerns are specifically related to the nature of information, therefore they fall outside of the scope of the ontological level. First, information has to be communicated and stored through the usage of symbols. Second, information is related to knowledge limitations; information may be false, incomplete, inaccurate, etc. Our capacity to store, transmit and process information is limited as well. Those limitations restrict the way we structure information about phenomena in reality. Finally, every information system may have particular ways of dealing with the various aspects of information handling, e.g., acquirement, communication, relevance and accuracy. Ergo, during the development of information systems, addressing those informational concerns is an unavoidable step.

1.2 GOALS

We have observed that ontology-based approaches and information modeling approaches have been treated in isolation, with the consequence that the relation between both is still in need of clarification. Our goal is to provide means to connect both levels in a systematic way. Our initial effort is to characterize both levels, on the lines of existing work on the ontological level, e.g., (Gruber, 1995; Guarino, 1994; Guizzardi, 2005), and on the information level, e.g., (Ashenhurst, 1996; Floridi, 2010; Halpin & Morgan, 2008; Kent, 2000; Langefors, 1980; Simsion & Witt, 2005). Specific care is taken to characterize the information level in a way that it can be appropriately related to the ontological level. We do such by identifying various concerns that are specifically related to the nature of information and investigate how they are related to ontological aspects. This process is aimed towards a two-level model-driven approach for conceptual modeling, with the purpose of

systematically developing a specification at the information level based on a specification at the ontological level. In sum, this thesis aims at addressing the following specific goals:

- To clarify the distinctions between the ontological level and the information level;
- To characterize the information level in conformance with the ontological level;
- To identify and characterize informational concerns and to relate those concerns with ontological aspects; and,
- To propose and implement a systematic transformation from an ontology-based conceptual model to an information model.

1.3 APPROACH[1]

We initially provide a metaphysical and terminological ground to be used henceforth in this thesis. Afterwards, we describe some basic notions underlying the ontological level and the information level. Finally, we briefly discuss how we systematically relate both levels by means of informational concerns and a model-driven approach to operationalize the creation of information models from ontology-based conceptual models.

1.3.1 REALITY, THOUGHTS AND SYMBOLS

In order to link both levels, we must lay a common ground to be shared by both. This ground is *reality*. Bunge, while establishing the thesis that "all science presupposes some metaphysics" (Bunge, 1977), provides a list of ten ontological principles occurring in scientific research. Here, we could use some of them, namely:

- "*There is a world external to the cognitive subject*";
- "*The world is composed of things*. Consequently the sciences of reality (natural or social) study things, their properties and changes";
- "*Things are grouped into systems* or aggregates of interacting components. (...) What there really is, are systems – physical, chemical, living, or social";
- "*Every system, except the universe, interacts with other systems in certain respects and is isolated from other systems in other respects*. (...) if there were no relative isolation we would be forced to know the whole before knowing any of its parts";
- "*There are several levels of organization*: physical, chemical, biological, social, technological, etc. The so-called higher levels emerge from other levels in the course of processes; but, once formed – with laws of their own – they enjoy a certain stability. Otherwise we would

[1] This section builds up on our earlier work on this theme that was reported in (Carraretto & Almeida, 2012).

> know nothing about organisms and societies before having exhausted physics and chemistry—which are inexhaustible anyway";

That is to say, there is an external world (*reality*) and there are levels of organization (*domains*), each with laws of their own. Domains could also be more specific, e.g., genealogy, engineering, business, law, medicine, archeology, molecular biology. "Domain" is also often referred to as "material domain", "subject domain", "universe of discourse" and "portion of reality".

We articulate about phenomena in reality in terms of concepts (e.g., matter and motion, chemical reactions, living organisms, social beings). As an illustration, a phenomenon could be articulated as: a certain portion of matter moving at a certain speed; a member of *Canis lupus familiaris* behaving outside its ecological niche; or Julia taking her puppy Toby for a walk. Which concepts should be adopted in articulations depends on criteria of abstraction, which are in part based in the domain being considered. Concepts and articulations are mental objects (*thoughts*); in order to communicate them, we assume the use of *symbols*.

Given those ideas, we can outline our own version of the so-called *triangle of reference*, illustrated in Figure 1.1. The original proposal (Ogden & Richards, 1923) illustrates that *words* and *things* are related only in an indirect manner by means of *thoughts*. In our terminology, the relation between reality (originally called "things") and symbols (originally called "words") is established by means of thoughts. We use the term "symbol" instead of "words" as we later refer to symbolic artifacts such as conceptual models and data. Henceforth in this thesis, we adopt the icons of Figure 1.1 whenever we apply the distinction between reality, thoughts and symbols.

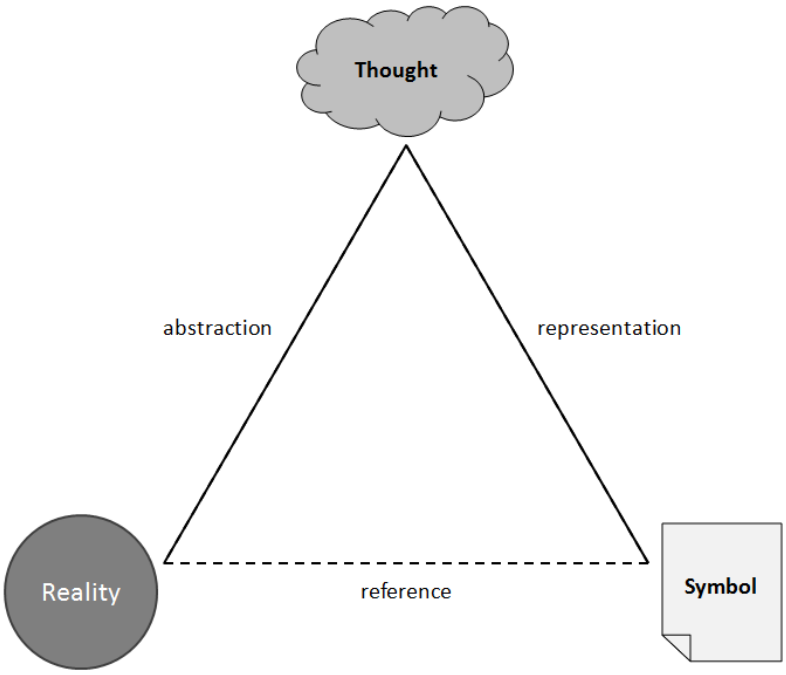

Figure 1.1 - Reality, thoughts and symbols

1.3.2 The Ontological Level

In the following, we characterize the basic elements of the ontological level, in line with (Guizzardi, 2005) and (Carraretto, 2010) with adapted definitions and terminology.

A *domain abstraction* is an articulation about phenomena in reality in terms of concepts (e.g., a man named John is the father of another man named Paul). A *domain conceptualization* is the set of concepts used to articulate domain abstractions (e.g., the concepts of Person, Man, Woman, Father, Mother, Offspring, being the father of, being the mother of). A domain conceptualization determines what kinds of things are considered to exist in the viewpoint of the user (or community thereof) that adopts it. That is to say, a domain conceptualization determines the "phenomena of interest" or "portion of reality". Domain conceptualizations and domain abstractions are abstract entities that only exist in the thought realm.

A domain conceptualization determines all *possible* domain abstractions that represent *admissible* phenomena in reality. For instance, in a domain conceptualization about genealogy, there cannot be a domain abstraction in which a person is his own biological parent. Given a domain conceptualization, there is only one *obtaining domain abstraction*, which describes reality in a perfect manner in terms of the domain conceptualization.

The relations between reality, domain conceptualizations and domain abstractions are summarized in Figure 1.2. First, reality can be seen according to different domain conceptualizations; each domain conceptualization determines a "portion of reality" that is of interest to a certain domain. Here, the relation between reality and a domain conceptualization is called "abstraction". For instance, in Figure 1.2, one domain conceptualization could abstract reality in terms of chemistry (thus, providing concepts such as Atom, Molecule and Covalent bond) and the other in terms of social contracts (thus, providing concepts such as Person, Organization and Employment). Each domain conceptualization is a means to articulate about "phenomena of interest" in reality through domain abstractions. Then, we say a domain abstraction is in "accordance" to a domain conceptualization and is an "articulation" about admissible phenomena in reality. For instance, given the chemistry domain conceptualization, a domain abstraction articulates about specific atoms and specific covalent bonds; and given the social domain conceptualization, another domain abstraction articulates about specific people, organizations and employments. In addition, the amount of detail provided by domain abstractions is always maximal given a domain conceptualization. This means, for example, that an obtaining domain abstraction about the social domain concerns *all about all* people, organizations and employments (e.g., existence, timing aspects, properties).

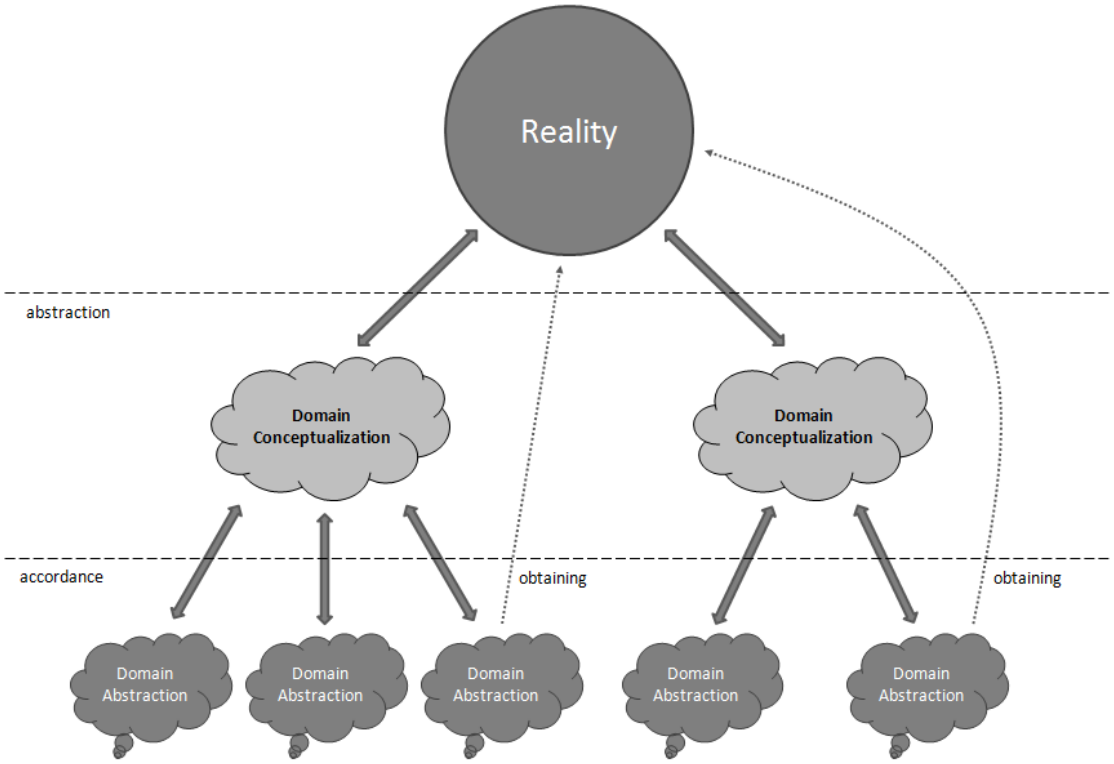

Figure 1.2 - Reality, domain conceptualizations and domain abstractions

In order to be documented, communicated and analyzed, domain conceptualizations must be captured in terms of some specification written in some *language*. Generally speaking, a *conceptual model* is the specification corresponding to a domain conceptualization and is written in a *general conceptual modeling language*. Authors such as Guarino (Guarino, 1994) and Guizzardi (Guizzardi, 2005) have defended that a conceptual modeling language should be rooted in a set of domain-independent concepts from a *foundational ontology* (a meta-conceptualization). A foundational ontology deals with formal aspects of entities irrespective of their particular nature, e.g., identity and unity, types and instantiation, rigidity, mereology, dependence (Guizzardi, 2005). In addition, a foundational ontology is philosophically and cognitively well-founded.

According to the advocates of ontological approaches, when conceptual modeling languages take into account formal distinctions, the potential misunderstandings and inconsistencies in conceptual models are reduced. That is to say, when ontological concerns are addressed, the quality of conceptual models is improved, facilitating thus the understanding and communication about a domain. This position characterizes the ontological level. At this level, conceptual models are written in an *ontology representation language*, i.e., a general conceptual modeling language that is rooted on a foundational ontology. Furthermore, a conceptual model written in an ontology representation language is called a *domain ontology*, which is the ultimate specification at the ontological level.

Figure 1.3 depicts the concepts discussed so far, which are fundamental for describing the ontological level.

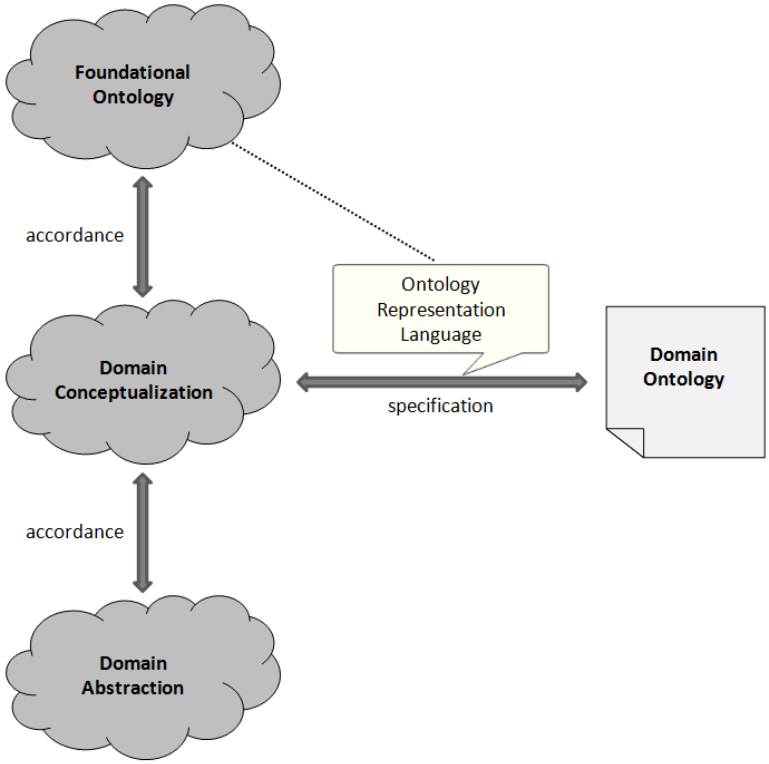

Figure 1.3 - Foundational ontology, ontology representation language and domain ontology

Finally, Figure 1.4 illustrates an example of domain ontology for genealogy, represented in a particular ontology representation language called OntoUML, originally proposed in (Guizzardi, 2005). This domain ontology provides the foundation for the parenthood relation between a mother (role played by a woman), a father (role played by a man) and an offspring (role played by a person). This model, by means of constructs and constraints, defines all the possible domain abstractions that are admitted by the conceptualization. In particular, this model determines that every person has a biological father and a biological mother independently of the availability of information regarding biological parents of a particular person.

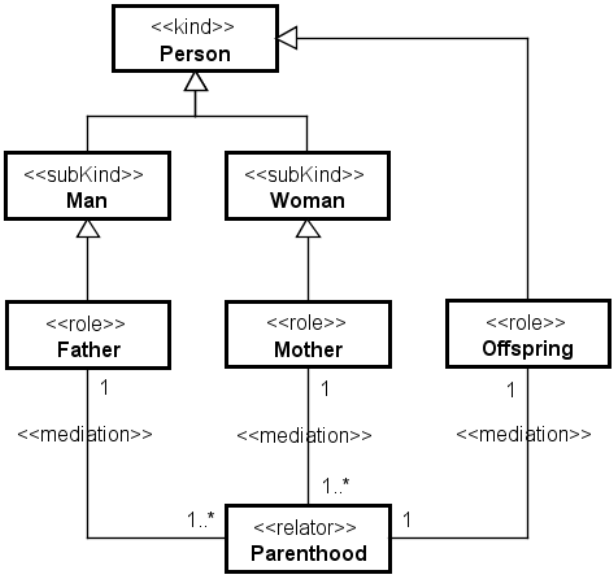

Figure 1.4 - A domain ontology about genealogy, written in OntoUML

1.3.3 The Information Level

The key element of the information level is *information*. Foremost, we consider information as an abstract thing (thought realm) that may be encoded using symbols. For instance, the sentence "John is married to Mary" conveys the information that the individual John is married to the individual Mary.

We assume that information is manipulated by agents that are fallible, i.e., capable of making mistakes or being erroneous. More specifically, we call those *informational agents*, which include human and artificial intelligence agents, as well as information systems. At the information level, human and artificial intelligence agents believe in pieces of information. Correspondingly, an information system manipulates pieces of *data* (symbol realm) that may be interpreted by its users in order to extract pieces of information from it. In this case, the information stored in the system (by means of data) is believed by its users. Generally speaking, we say that pieces of information are the objects of belief of informational agents.

In addition, we restrict our usage of the term "information" to *factual semantic content,* i.e., information that is intended (but may fail) to describe phenomena in reality. As a result, a piece of information may either be true or false. A piece of information is true if there is a phenomenon in reality corresponding to the informed semantic content; and false otherwise.

We assume that an informational agent adopts a domain conceptualization to which the manipulated pieces of information conform to. That is to say, a domain conceptualization is part of the beliefs of an informational agent, acting as a "semantic background" for the beliefs about phenomena in reality (i.e., pieces of information). As a consequence, in order to exchange information, agents must share a common domain conceptualization.

At the information level, an informational agent's knowledge about reality is potentially limited. Given a domain conceptualization, reality may be described in a potentially partial manner by means of pieces of information. For example, consider a domain conceptualization describing marriages as social contracts between a husband and a wife. First, given a domain conceptualization, pieces of information may describe the same phenomenon in reality with different amounts of detail, e.g., "John married to Mary in 1979-04-08", "John is married to Mary", "John is married", "Mary is married", "John is married since 1979". Furthermore, some phenomena in reality may be unknown to an informational agent. For instance, an agent may not know about past marriages (divorce or deceased spouse) or may not know about men, just women. In addition, as previously mentioned, pieces of information may actually be false and not correspond to any phenomena in reality. Hence, an agent's knowledge of reality may be limited and wrong.

Given the aforementioned aspects, we can contrast the information level with the ontological level. If there is agreement at the ontological level, then there is a unique view of reality given a domain conceptualization, namely, the obtaining domain abstraction. Distinctively, at the information level, informational agents, when given the same domain conceptualization, may have different knowledge of reality by means of pieces of information (beliefs). Figure 1.5 illustrates some of the distinctions between the ontological level and the information level by contrasting an obtaining domain abstraction with an informational agent's knowledge about phenomena in reality (given a domain conceptualization).

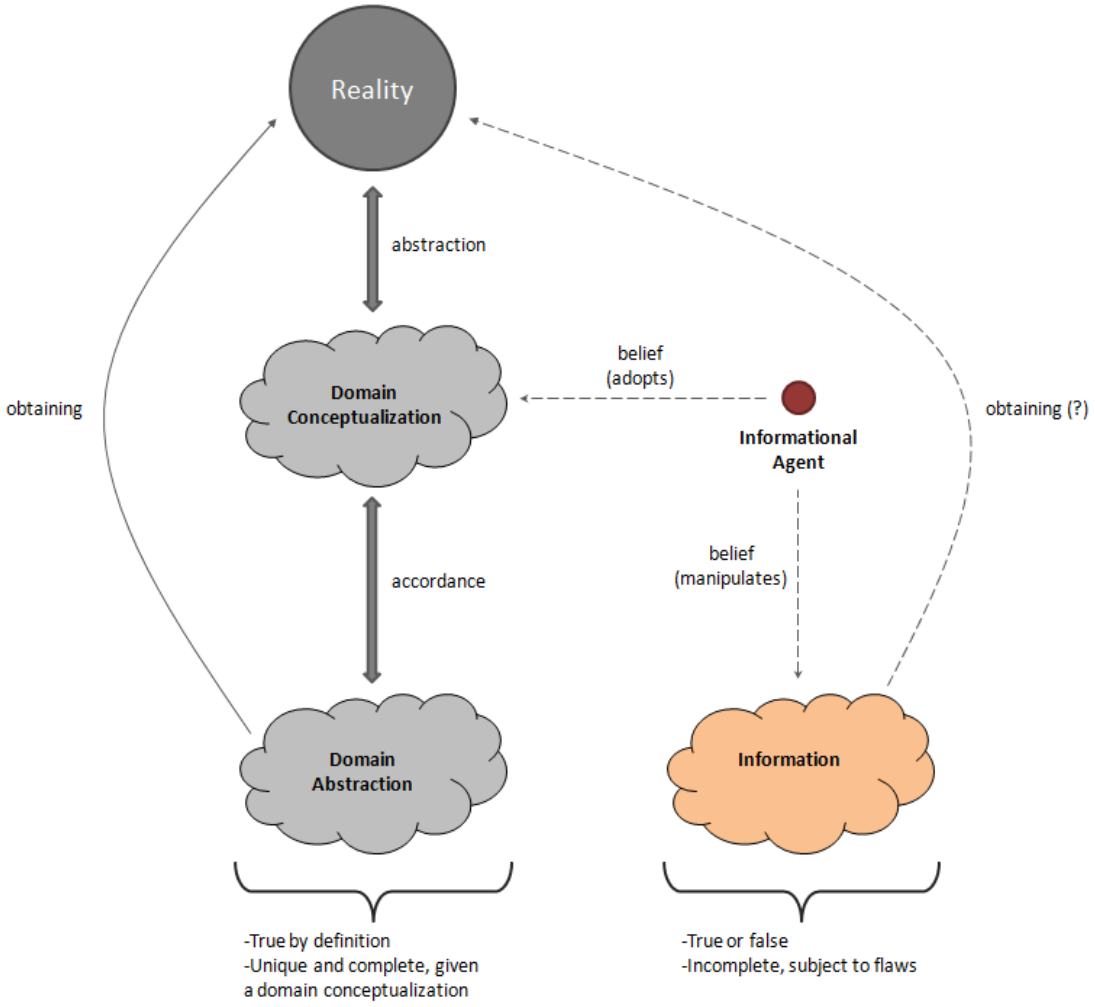

Figure 1.5 - An obtaining domain abstraction and an agent's knowledge

We consider that an informational agent has an *information demand*, which determines what the agent is required to know about phenomena of interest. Although informational agents may share a domain conceptualization, each agent potentially has a different information demand. That is to say, information manipulation, according to a domain conceptualization, inevitably requires the addressing of an information demand, which may vary from agent to agent. In order to address an information demand, an agent manipulates (and ultimately "embodies") data that conforms to what we call here an *information structure*.

An information structure only captures abstract *syntactical* structures, i.e., it only specifies how pieces of symbolic data are formed, but does not specify pieces of information as beliefs. Information as belief must be extracted via interpretation of well-formed data, i.e., data that conforms to an information structure. Data interpretation involves a number of underlying premises about what the data is supposed to mean. Those premises are not part of the information structure, but they must be known by the interpreter in order to completely extract information from data. Finally, an information structure is captured by means of the ultimate specification at the

information level, namely, an *information model*, which is written in an information modeling language. An information model is also called a conceptual schema, and it is usually expressed in ER diagrams (Chen, 1976), UML class diagrams (Rumbaugh, Jacobson, & Booch, 1999) and ORM fact-based models (Halpin & Morgan, 2008). Figure 1.6 depicts the important concepts at the information level.[2]

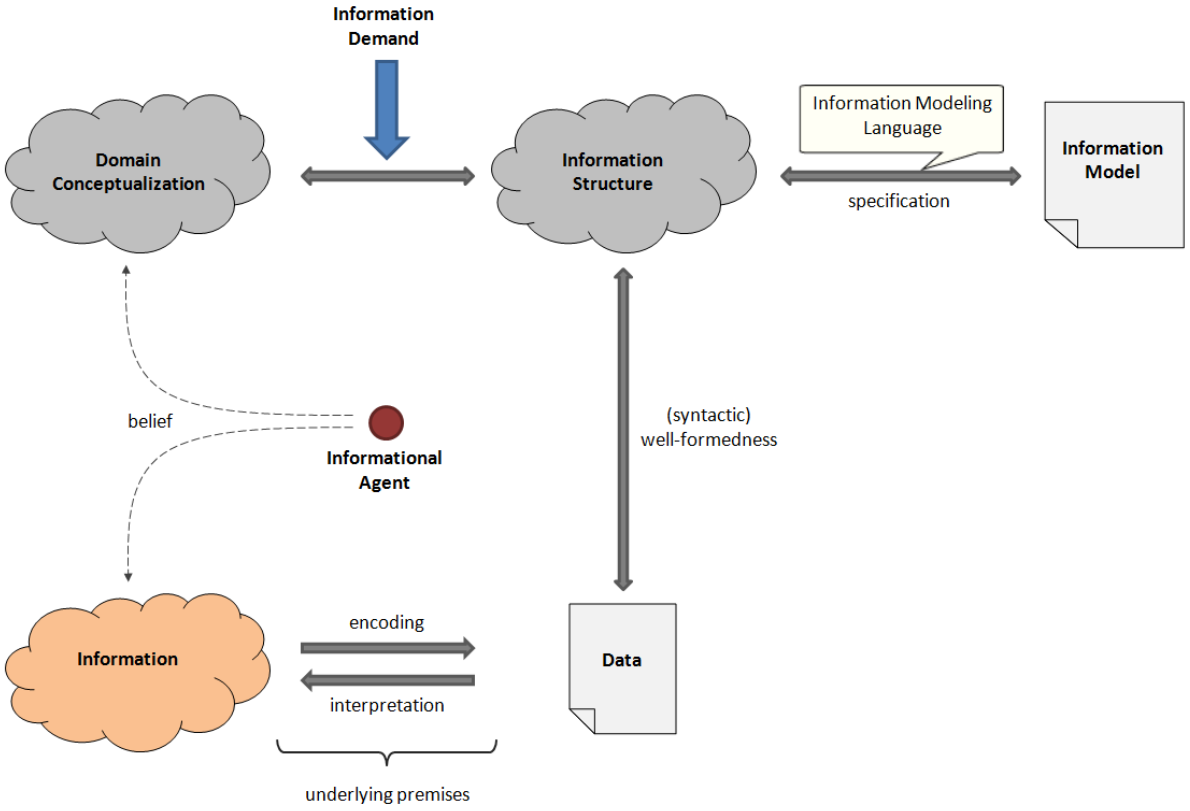

Figure 1.6 - Information demand, information structure and information model

To clarify the difference between a domain conceptualization and an information structure, we could examine the ontological level and the information level in terms of constraints. Constraints have different meanings at each level. At the ontological level, constraints are used to rule out inadmissible phenomena in reality, while at the information level, constraints settle the syntax of well-formed data, potentially taking into account the ignorance of the informational agent about reality. For example, consider a domain conceptualization for genealogy, capturing a parenthood relation between siblings and their parents. Let us assume that, from a biological perspective, all persons are siblings and parenthood must be established between a sibling and both a mother and a father (one's biological parents). On the one hand, constraints in a domain conceptualization would

[2] We acknowledge that what is called by the prevailing literature as "information modeling" is actually a misnomer, since it only describes the syntax of data. Thus, what we called "information structure" and "information model" would be better called "conceptual data structure" and "conceptual data model".

reflect the real-world constraint (from the domain of genealogy) and require both a mother and a father for each person. A domain abstraction with a child that has no parents (biologically speaking) is simply deemed inadmissible by the domain conceptualization. On the other hand, an information structure for this domain conceptualization reflects whether one is required to *know* both parents, and thus may relax constraints with respect to a corresponding parenthood relation. It is perfectly possible that an informational agent has information on the child and no information on the parents; or yet, that it knows only one of the parents.

1.3.4 Informational Concerns

At the information level, we consider that the addressing of an information demand involves the addressing of different *informational concerns*. Among the concerns we identify here are: scope, history tracking, time tracking, reference and measurement. Furthermore, we aim to provide a systematic way of dealing with each informational concern by means of *informational decisions*. The addressing of an information demand via concerns and decisions should guide the construction of information models. In the following, we briefly discuss informational concerns and some of the corresponding informational decisions.

What we refer to as *scope* concerns the selection of concepts, from the domain conceptualization, whose instances are relevant to keep track of. For instance, consider a domain conceptualization defining the concepts of Person, Man and Woman. One agent may be interested in storing data about people, without applying the distinction between men and women. Another agent may be exclusively interested on data about women. As another illustration, consider a domain conceptualization defining the concepts of Employment, Employee and Employer. An agent may be exclusively interested on knowing how many employments an organization has, unconcerned about employees; other agent may be only interested on knowing whether a person is employed or not, unconcerned about the employer.

At the information level, concerns about what is relevant to remember arise. Thus, we identify informational decisions involving *history tracking*. For example, consider the domain of organizations and employees. For human resource management, one may be concerned with both current and past employments; for pay roll, one may be interested exclusively on current employments. In another example, consider the domain of persons and their weights. If one is interested in information for controlling weight loss, one may be interested in the history of values assumed by a person's weight; in the context of a boxing championship, one may simply be interested in the latest weight value measured in a predefined point in time; in the context of drug administration, one may simply be interested in the current weight value in order to calculate the appropriate dosage.

We identify informational decisions regarding *time tracking*, i.e., knowledge of timing aspects of things, which are often implicit in domain ontologies and are addressed by the theories of the ontological level. These decisions determine whether agents require to know about the timing of relations and of lifecycles of entities in the domain. Consider, for example, a domain about allocation of resources to projects. In the domain ontology, when allocations are treated as foundations for relations between resources and projects, allocations will possess underlying aspects such as start time, end time and duration (as a direct consequence of being categorized as foundations for relations). Nonetheless, which of those aspects are part of the information demand is an agent-specific decision. Some agents may be interested when resources were allocated and deallocated (i.e., interest in both start time and end time), while others may be interested in how long the allocations lasted (i.e., sole interest in the duration).

There are also informational decisions concerning *reference*. More specifically, at the information level, one must formalize how agents refer to entities in reality, through symbols that we call *identifiers*. For instance, agents may refer to people using different types of identifiers such as names, national identification numbers or arbitrary internal codes. Moreover, agents may be concerned about aspects related to the origin of identifiers, e.g., when an identifier was attributed to a certain individual and who made such attribution.

Finally, we identify informational decisions concerning *measurement*, which are related to considering the abstract data types adopted for measured values (unit dimensions, granularity). As an illustration, there are multiple ways of measuring height, e.g., using a ruler with centimeter precision or a measuring tape with 1/32 inch precision. Besides that, agents may be concerned about other qualitative aspects of measured values, e.g., when the value was obtained, who performed the measurement, what was the measuring instrument, what were the environmental conditions.

To illustrate about informational concerns, we present an example of information model in Figure 1.7. The figure shows an ORM model addressing the demand to maintain information on the history of patient weights, where for each patient at most one weight measurement is performed per day. All the five aforementioned informational concerns are addressed in this model. Concerning scope, the model provides constructs for data on patients (the Patient type), but not for other categories that a patient might depend on, e.g., the treatment and the doctor. Concerning reference, the model determines that each patient is identified by a patient number (see the reference mode "(.nr)" inside the rounded rectangle of the Patient type). Concerning measurement, the model determines that Weight is to be measured in Kilograms. Concerning history tracking, in order to store weight measurement records, the model includes a "WeightMeasurement" type. Concerning time tracking, the model explicitly represents a construct for dates of weight measurements (the Date

type). Also, it defines that those dates are represented in the year-month-day format, specifying thus a particular granularity of time (a measurement decision).

Figure 1.8 and Figure 1.9 depict, respectively, the corresponding UML model and ER diagram for the same information demand. The UML and ER versions are slightly different from the ORM one in terms of structure and constraint specification. In UML and ER, Date and Weight are represented as attributes instead of types. Also, in the UML model, the constraint that guarantees that each patient is measured only once per day has to be declared separately.

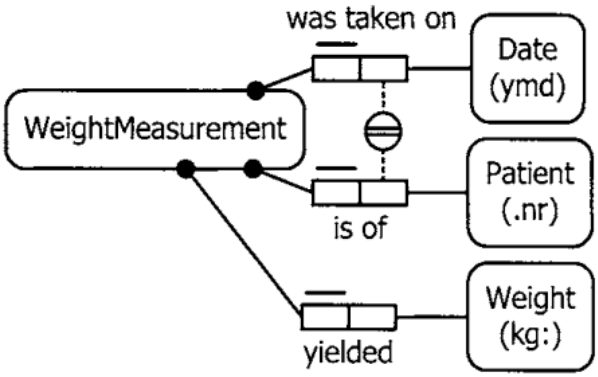

Figure 1.7 - Informational concerns addressed in ORM (taken from (Halpin & Morgan, 2008))

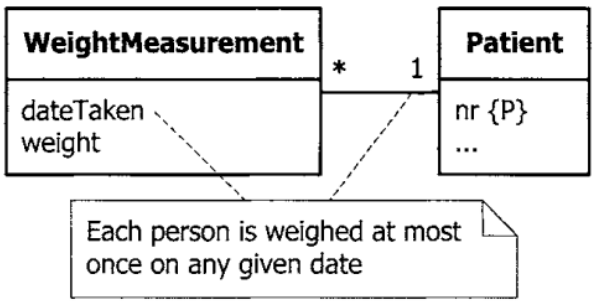

Figure 1.8 - Informational concerns addressed in UML (taken from (Halpin & Morgan, 2008))

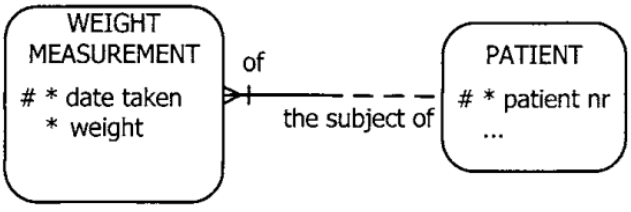

Figure 1.9 - Informational concerns addressed in ER (taken from (Halpin & Morgan, 2008))

1.3.5 Model-driven Approach and Tool Support

In order to address an information demand over a domain conceptualization, we propose a model-driven approach for conceptual modeling, with the purpose of systematically developing an information model (at the information level) based on a domain ontology (at the ontological level), in a process that is guided by the identified informational concerns and decisions. In our model-driven

approach, we use OntoUML (Guizzardi, 2005) as the source language for domain ontologies and UML as the target language for information models.

In our approach, we consider that an information model could play an important role during the *design* phase of information systems development. Therefore, the resulting information model commits to certain *design decisions* that could guide a further *implementation* phase. Hence, by means of an information model, we aim to bridge the gap between a domain ontology and an implementation (or logical) model. The resulting information model is a specification about *abstract data types* (instead of implementation-related constructs, such as database tables or programming language classes).

In order to show the feasibility of the model-driven approach, we implement the transformation in an integrated development environment (IDE), namely, Eclipse, by applying metamodeling technologies such as the Eclipse Modeling Framework (EMF)[3] and the Ecore metamodeling language. The transformation is integrated as a component of the OntoUML Modeling Infrastructure (Carraretto, 2010).

1.4 STRUCTURE

The remainder of this thesis is structured as follows (Figure 1.10):

- Chapter 2 (The Ontological Level) presents important background concepts for the ontological level. Those concepts will help contrast the ontological level with the information level and will be further used to systematically address informational decisions. We describe a portion of the Unified Foundational Ontology (UFO) and the ontology representation language that is based on it, a UML profile for class diagrams called OntoUML.
- Chapter 3 (The Information Level) presents an in-depth characterization of the information level. We provide an investigation about the notions of data and information, and we clarify our notions of informational concerns, informational decisions and information models.
- Chapter 4 (From a Domain Ontology to an Object-Oriented Information Model) provides the first step in our model-driven approach, as it presents the patterns for generating the basic structure of an object-oriented information model based on the structure of a domain ontology.
- Chapters 5 (Scope), 6 (History and Time tracking) and 7 (Reference and Measurement) discuss the corresponding informational concerns and present the model-driven informational decisions for addressing them.

[3] http://www.eclipse.org/modeling/emf/

- Chapter 8 (Tool support) describes our tool support for the model transformation from OntoUML to UML in the Eclipse platform.
- Chapter 9 (Related Work) discusses the available work in the literature about the separation of concerns in conceptual modeling.
- Chapter 10 (Conclusions) outlines the main contributions of this thesis and proposes topics for further investigation.

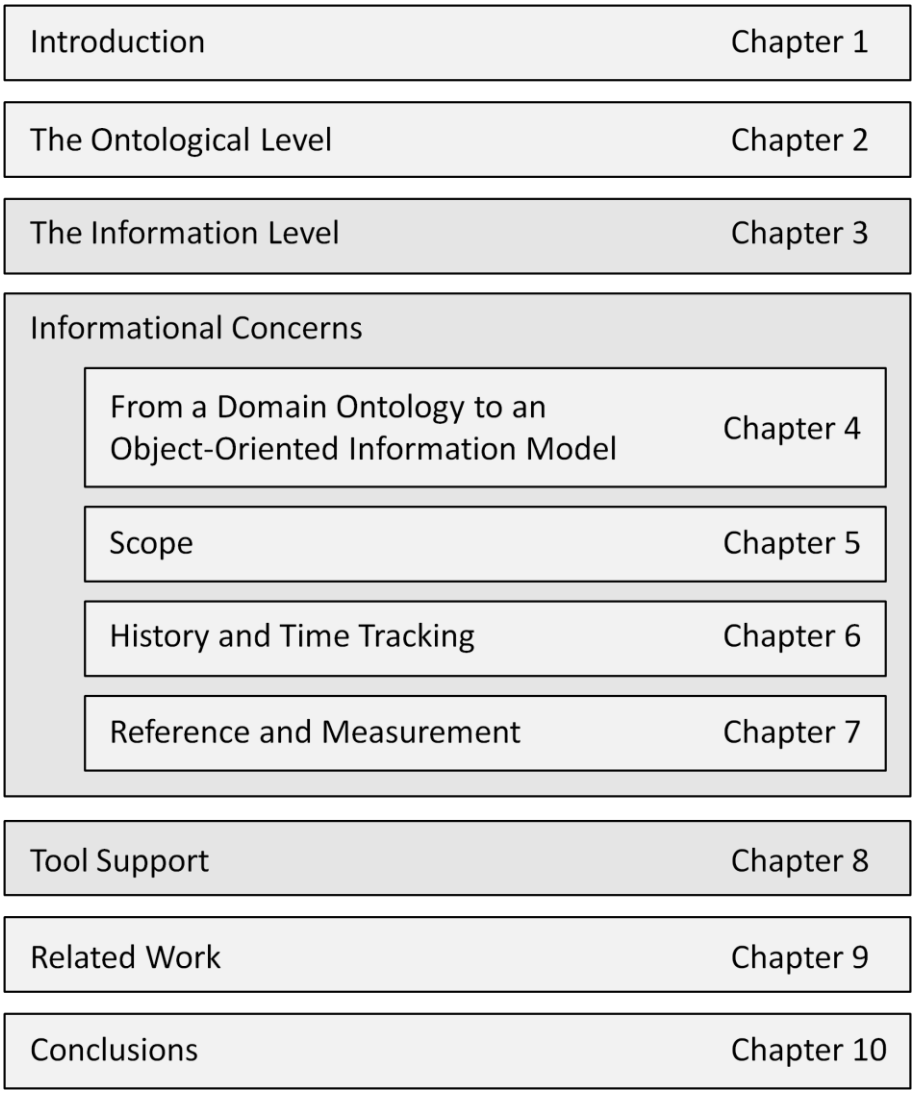

Figure 1.10 - Thesis structure

2 THE ONTOLOGICAL LEVEL

Differently from the information level, the ontological level concerns the nature of phenomena of interest, addressing the categories of being which are assumed to exist in a certain domain independently of particular information demands. Sections 2.2, 2.3 and 2.4 are heavily based on (Guizzardi, 2005) and, therefore, quotation marks will be omitted.

2.1 UFO AND ONTOUML

In order to characterize the ontological level and to help us articulate about informational decisions, we adopt here some concepts from a foundational ontology, namely, the Unified Foundational Ontology (UFO) (Guizzardi, 2005). UFO is a domain-independent system of categories dealing with formal aspects of objects, addressing ontological aspects such as identity and unity, types and instantiation, rigidity, mereology and so forth. It has been developed from a combination of the GFO (Generalized Formal Ontology) underlying GOL (General Ontology Language) (Heller & Herre, 2004) and the Ontology of Universals underlying OntoClean (Guarino & Welty, 2002). We describe here only the top-level concepts which are relevant to the scope of this thesis.

Besides the theoretical work, the approach defended on Guizzardi's thesis was instantiated by proposing a conceptual modeling language that incorporates the foundations captured in UFO. The Unified Modeling Language (UML) (OMG, 2011) was analyzed and redesigned with the objective of proposing an ontologically well-founded version of it that can be used as an appropriate language for the ontological level. This proposed extension of UML is called OntoUML, which we adopt here as a language for specifying domain ontologies. In this chapter, we describe the portions of OntoUML that are relevant to this thesis. The relation between UFO (a meta-conceptualization), OntoUML (a language) and OntoUML models (domain ontologies) is depicted in Figure 2.1.

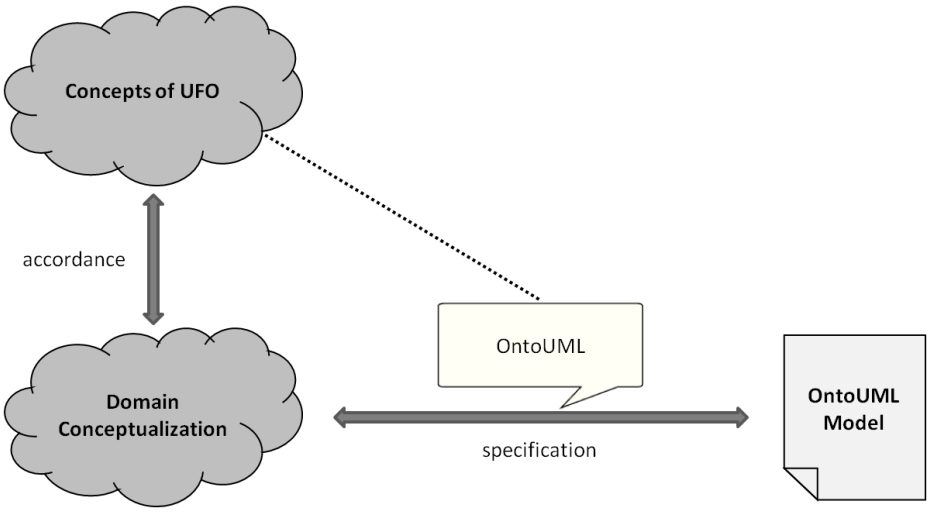

Figure 2.1 - UFO (meta-conceptualization) and OntoUML (language)

2.2 UNIVERSALS AND INDIVIDUALS

2.2.1 UFO

The first fundamental distinction adopted in UFO is that between universals and individuals. Universals are patterns of features that can be realized in a number of different individuals. For example, in some situation, Mary is an individual (instance) of the universals Person, Woman and Wife, while John is an individual (instance) of the universals Person, Man, Husband and Employee.

Individuals can be distinguished in terms of their behavior w.r.t. time. Endurants are said to be wholly present whenever they are present, i.e., they are in time (e.g., a person, the brightness of the Sun). Perdurants (henceforth, events) are individuals composed of temporal parts, i.e., they happen in time (e.g., a birth, a marriage ceremony, a race). Endurants can be divided into substantials and moments. Substantials are existentially independent individuals (e.g., a person, a forest, a lump of clay). Moments, in contrast, are individuals that can only exist in other individuals, and thus they are existentially dependent on other individuals (e.g., a table's height, a person's headache, a covalent bond between atoms).

For every category of individuals discussed so far, there is a corresponding category of universals. For instance, substantial universal is the category of universals whose individuals are substantial individuals. Thus, the categories of universals uncovered so far are substantial universals, moment universals, endurant universals and perdurant universals. As an illustration, hitherto, a Person universal would belong to the categories of substantial universal and endurant universal, while a Height universal would belong to the categories of moment universal and endurant universal. There are further categories of universals into which individuals can be classified throughout their lifecycle; we describe them in the following.

The first distinction among universals is based on the notion of a principle of identity which supports the judgment whether two individuals are the same (i.e., in which circumstances the identity relation holds). Substantial universals that carry a principle of identity for the individuals they collect are called *sortal* universals (e.g., Apple). *Mixin* universals, on the other hand, are substantial universals that represent an abstraction of properties that are common to multiple disjoint types (e.g., Red Thing) and thus do not carry a principle of identity.

An important meta-property that is used to distinguish some universals is called *rigidity*. A universal is *rigid* if every instance of it is necessarily (in the modal sense) an instance of it. An example of rigid universal would be Person, since instances of Person cannot cease to be so without ceasing to exist. Conversely, a universal is *anti-rigid* if every instance of it is possibly (in the modal sense) not an instance of it. Student is an example of anti-rigid universal, since every instance of Student may cease to be so without ceasing to exist.

A Kind universal is the unique rigid sortal universal that provides a principle of identity for its individuals (e.g., Person). A Kind universal can be specialized in other rigid universals that inherit its supplied principle of identity; those universal are called *SubKind* universals (e.g., Man and Woman, which inherit their identity principle from Person). C*ategory* universals are rigid Mixin universals representing an abstraction of properties that rigidly apply to different Kinds. For example, Legal Entity is a Category abstracting the property of being subject to legal responsibilities, which is possessed by both Person and Organization Kinds.

Role universals are anti-rigid sortal universals that depend on extrinsic properties in the context of a relation (e.g., Student participating an Enrollment along with University, Husband and Wife participating a Marriage). *Role Mixin* universals are anti-rigid Mixin universals that represent abstractions of common properties of roles of different Kinds (e.g., Customer Role Mixin played by instances of Person and Organization Kinds). Both Role universals and Role Mixin universals are further described in section 2.4 (which discusses “role playing”).

The categories of individuals and universals that were previously discussed are represented in Figure 2.2 (the terms “universal” and “individual” have been abbreviated).

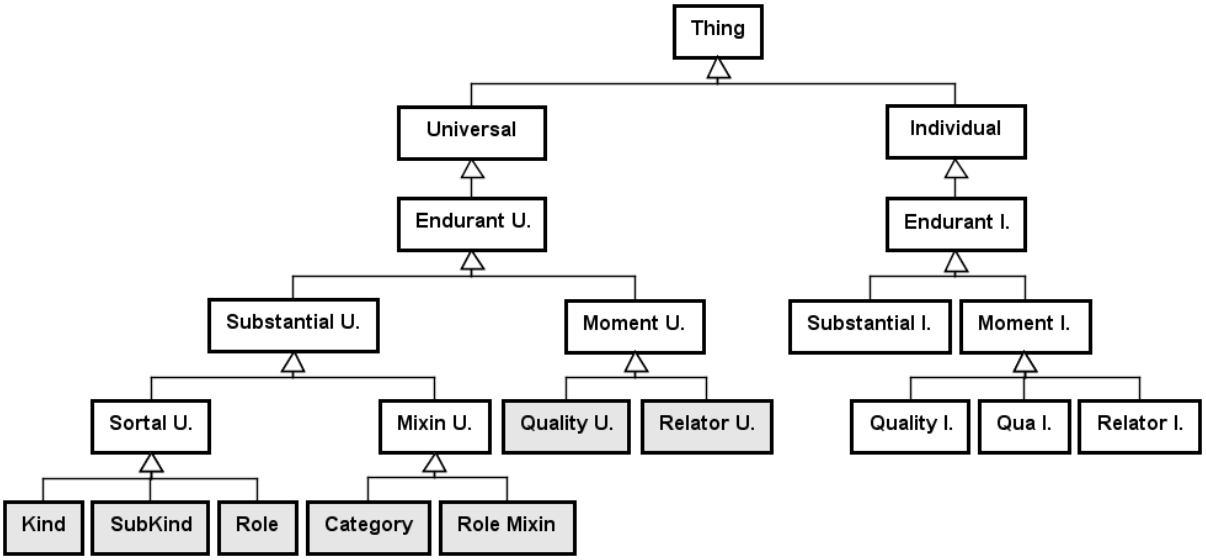

Figure 2.2 - Categories of universals and individuals

2.2.2 OntoUML

The meta-categories of UFO have driven the revision of UML by means of the OntoUML profile. More specifically, the OntoUML language incorporates constructs (UML stereotypes) that are based on some of the ontological distinctions provided by UFO. An OntoUML model (i.e., a domain ontology written in OntoUML) is a specification about universals (not individuals).

In Figure 2.3, we present an OntoUML domain ontology involving the rigid concepts of Kind, SubKind and Category universals. Each of those categories of universals is represented as UML

classes with a stereotype named after the category's name: <<kind>>, <<subKind>>, <<category>>, respectively. Mixin universals cannot have direct instances, therefore they are always represented as abstract UML classes (their names are italicized). Category universals, as sorts of Mixin universals, follow this constraint and this can be noticed in the Legal Entity Category in the domain ontology.

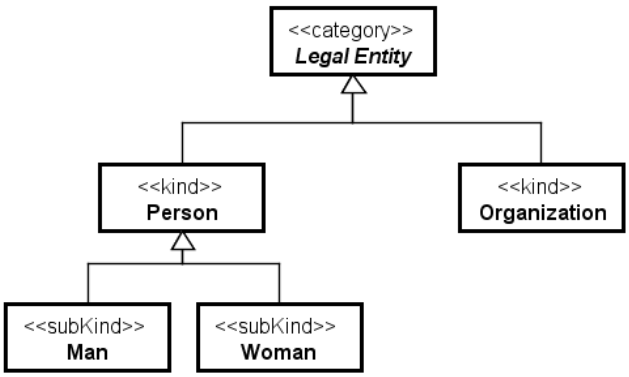

Figure 2.3 - Rigid universals in OntoUML

Universals in the domain ontology can be related via UML generalizations. Nonetheless, there are several constraints that must be obeyed. Some relevant constraints to be cited here are: (a) a Kind universal cannot specialize a Rigid Sortal; (b) a Rigid Sortal universal cannot specialize an Anti-Rigid universal; (c) A Mixin universal cannot specialize a Sortal universal.

2.3 MOMENTS AND QUALITIES

2.3.1 UFO

As previously mentioned, moments are individuals that are existentially dependent on other individuals. An individual *x* is existentially dependent on another individual *y* if, as a matter of necessity, *y* must exist whenever *x* exists. Existential dependence is a necessary but not a sufficient condition for something to be a moment. For instance, the temperature of a volume of gas depends on, but is not a moment of its pressure.

Thus, for an individual to be a moment of another individual, a relation of *inherence* must hold between the two. For example, inherence "glues" one's smile on one's face, or the charge in a specific conductor to the conductor itself. Therefore, inherence is a special type of existential dependence relation between individuals. Consequently, a moment is defined as an endurant that *inheres in* another endurant. The endurant in which a moment inheres is said to *bear* the moment, and is called its *bearer*.

As mentioned, inherence is a relation between individuals, namely, a moment individual and another individual, its bearer. The counterpart of inherence for universals is a relation between a universal and a moment universal, which is called *characterization*. A universal *is characterized by* a

moment universal if every instance of the former bears an instance of the latter. The moment universal is said to *characterize* the other universal, which is called the *characterized* universal.

Now, we focus our attention on a particular type of moment, namely, qualities. According to the DOLCE foundational ontology, qualities "can be seen as the basic entities we can perceive or measure" and they inhere in specific individuals ("no two [individuals] can have the same quality") (Gangemi, Guarino, Masolo, Oltramari, & Schneider, 2002). As a consequence DOLCE (and UFO alike) distinguishes "between a quality (e.g., color of a specific rose), and its 'value' (e.g., a particular shade of red)" (Gangemi et al., 2002). The latter is called quale, and describes the position of an individual quality within a certain quality structure. So "when we say that two roses have (exactly) the same color their two colors have the same position in the color space (they have the same color quale), but still the two roses have numerically distinct color qualities" (Gangemi et al., 2002).

Each *quality structure* is "endowed with certain geometrical structures" and is supposed "to satisfy certain structural constraints" (Gärdenfors, 2000). For example, for the weight quality, there is a quality structure "which is one-dimensional with a zero point and thus isomorphic to the half-line of nonnegative numbers" (Gärdenfors, 2000). As another example, "our cognitive representation of colors can be described by three dimensions: hue, chromaticness, and brightness" (Gärdenfors, 2000) (those dimensions form the quality structure of the color spindle). Finally, "there is, in general, no unique way of choosing a [quality structure] to represent a particular quality but a wide array of possibilities" (Gärdenfors, 2000).

To clarify this theory, consider the following example: suppose we have two substantials, *a* (a red apple) and *b* (a red car), and two qualities, q_1 (particular color of *a*) and q_2 (particular color of *b*). When saying that *a* and *b* have the same color, we mean that their individual color qualities q_1 and q_2 are (numerically) different, however, they can both be mapped to the same point in the color quality structure, i.e., they have the same quale. The relations between a substantial, one of its qualities and the associated quale are summarized in Figure 2.4.

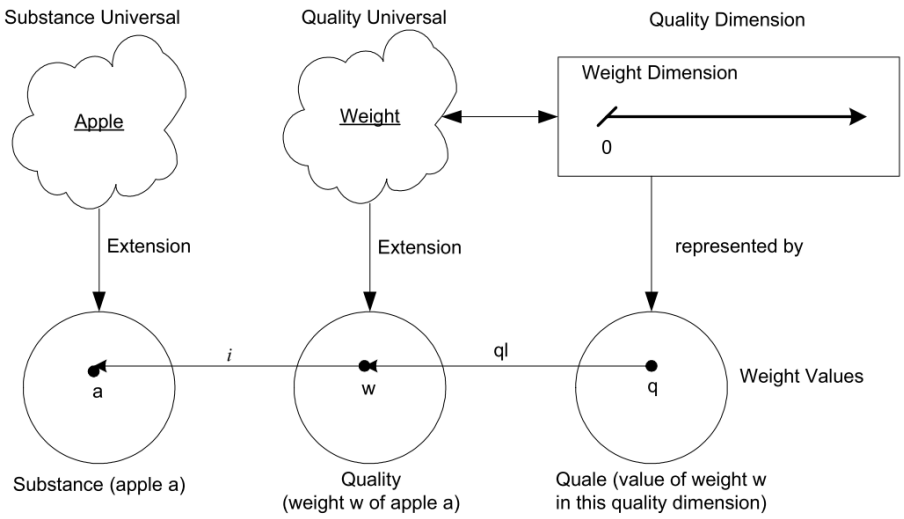

Figure 2.4 - Substantial, Quality and Quale (Guizzardi, 2005)

2.3.2 OntoUML

We adopt here a representation of Qualities in OntoUML that is different from the one proposed in (Guizzardi, 2005). We represent qualities as UML classes stereotyped as <<quality>> and connect them to their bearers via an association stereotyped <<characterization>>. This is depicted in Figure 2.5. In the original proposal by (Guizzardi, 2005), universals are directly connected to data types that represent the lexicalization of quality structures. That is to say, in Guizzardi's profile, Qualities are not explicitly represented in a domain ontology and data types are represented instead[4]. In our approach, we assume that users of a domain ontology should agree on the geometrical properties of a quality structure, but not necessarily on its lexicalization. Therefore, we leave the encoding of a quality structure into a data type to information models, as we discuss in chapter 7. This issue will become more evident as we elaborate on the distinctions between data and information and as we clarify the responsibilities of an information model in chapter 3.

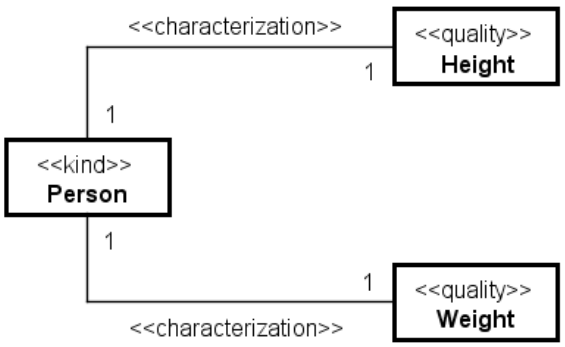

Figure 2.5 - Our representation of qualities in OntoUML

[4] Although the explicit representation of qualities is recognized, e.g., in (Guizzardi, Masolo, & Borgo, 2006).

There are several constraints involved in the explicit representation of Qualities in domain ontologies. First, a <<quality>> must be connected to a <<characterization>>. A <<characterization>> must be binary, the source end must be the characterizing moment (in our case, a quality universal), and the target end must be the characterized universal. Additionally, the source end (moment end) minimum cardinality must be greater or equal to 1, and the target end (bearer end) must be "read only" and its cardinality must be exactly 1. In a domain ontology, the categories of universals that can be related to a <<quality>> via <<characterization>> are <<kind>>, <<subKind>>, <<category>> and, as we describe in the following section, <<relator>>.

2.4 Role Playing

2.4.1 UFO

In this section, we provide ontological foundations on role playing situations. Substantials may play roles in the context of the so-called *material relations*, e.g., being employed in, being enrolled in, being married to. Material relations are said to alter the history of the involved substantials. This is a relevant criterion for distinguishing material relations from the so-called formal relations, e.g., being taller than, being heavier than, being older than. As an illustration, the individual histories of John and Mary are different because of the material relation of John being married to Mary, while the same is not true for the formal relation of John being taller than Mary.

Substantials in the context of material relations are said to be mediated by individuals called *relators*. Relators are individuals with the power of connecting substantials, e.g., an employment relator connects a person (an employee) and an organization (an employer), an enrollment relator connects a person (a student) and a university, a marriage relator connects two people (a husband and a wife). In the sequel, we define the idea of a qua individual, which will be important to characterize relators.

Consider the material relation of John being married to Mary. In the context of their marriage (as a social contract), there are many externally dependent moments of John that depend on the existence of Mary, e.g., all responsibilities that John acquires by virtue of this foundation. A *qua individual* is an individual that bears all the externally dependent moments of a substantial (e.g., John) that share the same dependencies (e.g., Mary) and the same foundation (e.g., marriage). Thus, a qua individual is a special type of externally dependent moment. Intuitively, a qua individual is the way a substantial participates in a certain material relation. The name comes from considering an individual w.r.t. certain aspects (e.g., John qua student, Mary qua musician).

As a result, a relator is defined as an aggregate of all qua individuals that share the same foundation. For example, for John being married to Mary, their marriage relator is the aggregate of

John qua husband of Mary and Mary qua wife of John. Qua individuals composing a relator are existentially dependent on each other. Furthermore, given the qua individuals that compose a relator, we say the relator *mediates* their bearers, the substantials. For instance, for John being married to Mary, their marriage relator mediates John and Mary. A relator must mediate at least two distinct substantials.

A *Relator universal* is a universal whose instances are relators. A mediation relation holds between a universal and a Relator universal if every instance of the former is mediated by an instance of the latter. A *Role universal* is a universal whose instances are substantials that bear a certain qua individual moment in the context of a material relation that is derived from a relator. Role universals are anti-rigid sortal universals, i.e., every instance of a Role universal is possibly not an instance of such role. A Role universal always has a mediation relation to a Relator universal.

A Material relation universal is a universal whose instances are material relations. We say a Material relation universal is *derived from* a Relator universal, or alternatively, we say a relation of *derivation* holds between a Material relation universal and a Relator universal. Individual material relations stand merely for the facts derived from the relator individual and its mediating entities. A relator individual is the actual instantiation of the corresponding relational property (the objectified relation).

Finally, a Role Mixin universal is an anti-rigid Mixin universal that addresses the problem of a role played by instances of different Kinds. As an illustration, consider a material relation of purchasing, between Customer and Supplier roles. In addition, consider that both people and organizations can be customers. In this case, Customer is a Role Mixin, i.e., it is an anti-rigid and relationally dependent universal that is played by instances of different Kinds.

2.4.2 OntoUML

The pattern for representing role playing in OntoUML is depicted in Figure 2.6. Role universals are represented by UML classes with the <<role>> stereotype, Relator universals by UML classes with the <<relator>> stereotype, and Mediation universals by UML associations with the <<mediation>> stereotype. Every <<role>> should be (directly or indirectly) connected to a <<mediation>>. Being indirectly connected means the <<role>> specializes another universal that is connected to a <<mediation>>. Analogously, every <<relator>> must be (directly or indirectly) connected to a <<mediation>>.

A <<mediation>> must be binary, its source end must be a <<relator>> and its target end must be the mediated universal. The source end (relator end) minimum cardinality must be greater or equal to 1. The target end (mediated end) must be “read only” and its minimum cardinality must be greater or equal to 1. Furthermore, for a <<relator>> and all its <<mediations>>, the sum of the

minimum cardinalities of the mediated ends must be greater or equal to 2. For example, in Figure 2.6, every instance of Employment mediates (at least) 1 instance of Employee and (at least) 1 instance of Employer, the sum in this case is (at least) 2.

Finally, in OntoUML, “the externally dependent moments of a qua individual are represented as resultant moments of the relator” (Guizzardi, 2005). As a consequence, Relators may be related to Qualities via Characterization, but not the mediated Roles and Role Mixins (the latter are discussed in the following).

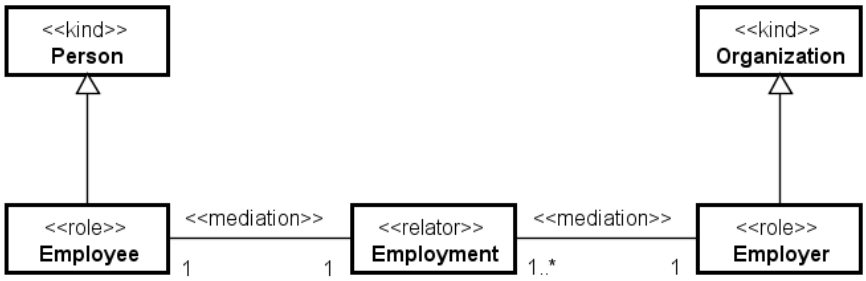

Figure 2.6 - Role playing pattern in OntoUML

The Role Mixin design pattern addresses how one should model the relationship between a Role Mixin and its allowed Kinds. For example, Customer has in its extension individuals that belong to different Kinds (viz. Person and Organization) and, thus, that obey different principles of identity. Hence, Customer is a Mixin and, by definition, cannot supply a principle of identity to its instances. Since an individual must obey one and only one principle of identity, every instance of Customer must be an instance of one of its sub-universals that, in turn, should carry that principle of identity. Then, we define the sortals Private Customer and Corporate Customer as sub-universals of Customer. These sortals, in turn, carry the (incompatible) principles of identity supplied by the Person and Organization Kinds, respectively. In summary, an instance of Customer (abstract class) must be an instance of exactly one of its sub-universals (e.g., Private Customer), which carries the principle of identity supplied by a Kind universal (e.g., Person).

An illustration of the Role Mixin pattern is depicted in Figure 2.7. A Role Mixin universal is represented as a UML Class stereotyped as <<roleMixin>>. As a Mixin universal, a <<roleMixin>> must be abstract. Also, a <<roleMixin>> must be (directly or indirectly) connected to a <<mediation>>. Finally, a specialization constraint involving <<roleMixin>> is the following: a Category (rigid Mixin) cannot specialize a Role Mixin (anti-rigid Mixin).

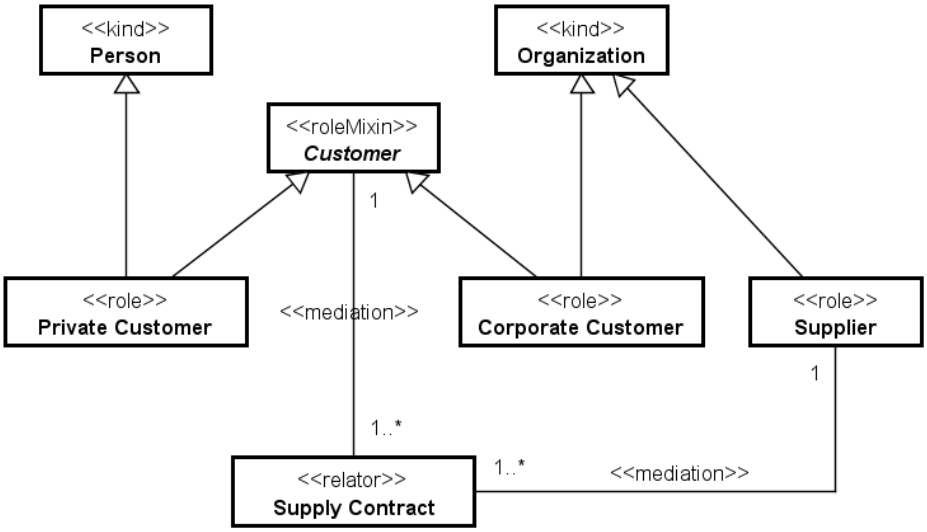

Figure 2.7 - Role Mixin pattern in OntoUML

2.5 Running Example

At last, in the remainder of this thesis, we illustrate our further explanations by means of a running example of domain ontology, depicted in Figure 2.8. This domain ontology is supposed to describe relations involving people and/or organizations (e.g., marriages, employments and supply contracts). In the following, we explain the underlying ontological commitments that we assume here.

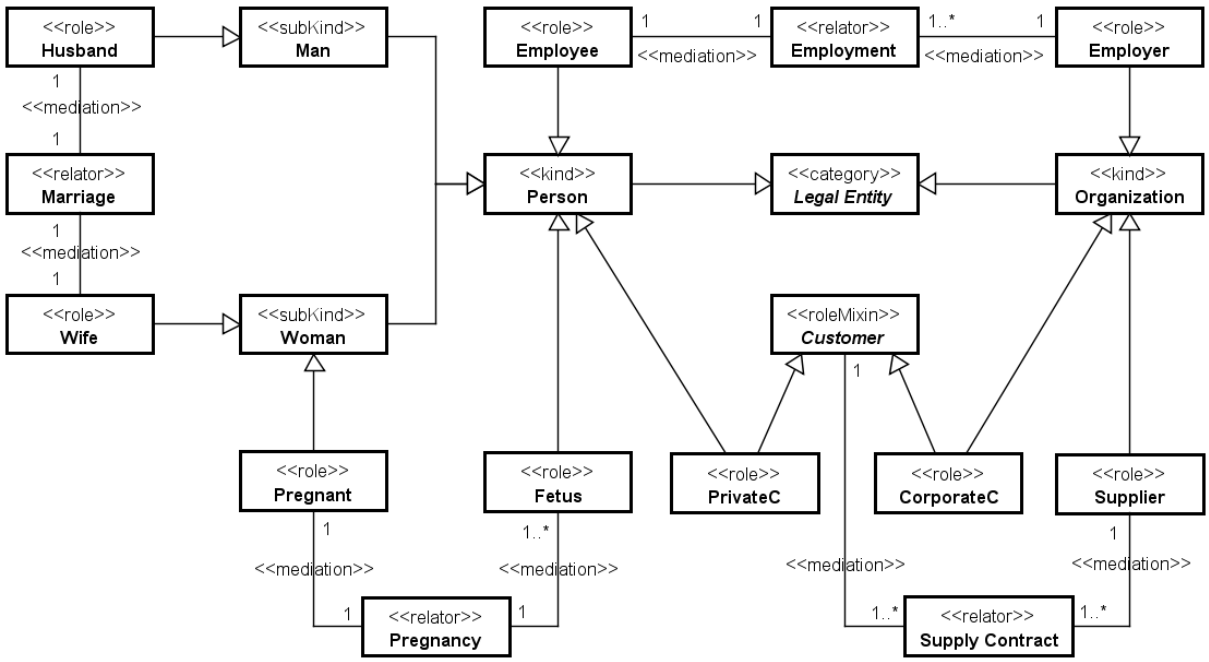

Figure 2.8 - Our running example of domain ontology

We assume a biological principle of identity for people, in which people begin to exist as fetus and cease to exist on death. For organizations, we assume the social principle of identity, in which organizations begin and cease to exist on social contracts (for startup and closure, respectively). For the sake of example, we assume both people and organizations are legally liable throughout their whole lifecycle (i.e., both natural and legal persons have legal obligations to pay debts).

We adopt the notion of marriage as a social contract, i.e., a marriage starts to exist on a marriage contract and ceases to exist on divorce or death. Similarly, we refer to employments as social contracts that start to exist on hiring and cease to exist on firing. The case is also analogous for supply contracts. Finally, we adopt the usual biological perspective on pregnancy, which ceases to exist on birth.

In the domain ontology, we specify the cardinality of relations with *current semantics*. For example, a man or a woman may only be related to one (existing) marriage contract at a time. Thus, we are left to describe, outside the model, the cardinalities of relations with *lifetime semantics*. We assume that, throughout the lifetime of entities (viz. people and organizations), all the specified relations (viz. marriages, pregnancies, employments and supply contracts) may be played without restriction.

3 THE INFORMATION LEVEL

In order to characterize the information level, we must properly define two core terms, namely, “data” and “information”. Those terms cannot be directly applied without some background discussion, since they are used in various areas (e.g., physics and chemistry, engineering, information theory, computer science) with different meanings, and sometimes are not properly defined. Once we clarify our usage of the terms “data” and “information”, we can establish the basic elements of the information level and thus clarify what we mean by informational concerns, informational decisions and information models. This sets the basic notions for the core approach presented in chapters 4, 5, 6 and 7.

3.1 DATA

Part of the definition of information relies on the definition of data. For instance, the General Definition of Information (GDI) is a popular way to define information as data plus meaning. According to the GDI, information (as semantic content) is made of data that is well-formed (syntax) and meaningful (semantics) (Floridi, 2010). Thus, we initiate our discussion with the definition of data.

An attempt to define “datum” (singular of “data”) in a way that can fit all its particular usages leads to a very general and vague definition of datum as “a lack of uniformity”. According to (Floridi, 2010), this “lack of uniformity” can be applied in three main ways:

- Lacks of uniformity in the real world (also called “data in the wild” or “dedomena”), i.e., whatever lack of uniformity in the world that looks to us as data (e.g., a red light against a dark background);
- Lacks of uniformity between (the perception of) at least two physical states of a system or signals (e.g., a higher or lower charge in a battery, a variable electrical signal in a telephone);
- Lacks of uniformity between two symbols (e.g., the letters B and P in the Latin alphabet);

Those three ways of seeing data are somehow related: “dedomena” may be either identical with (or what makes possible) signals, and signals are what make possible the coding of symbols. Furthermore, Floridi acknowledges:

> The dependence of information on the occurrence of syntactically well-formed data, and of data on the occurrence of differences variously implementable physically, explain why information can so easily be decoupled from its support. The actual format, medium, and language in which data, and hence information, are encoded is often irrelevant and disregardable. In particular, the same data/information may be printed on paper or viewed on a screen, codified in English or in some other language, expressed in symbols or pictures, be analogue or digital. (Floridi, 2010)

For our purposes, at the information level, we disregard *physical* aspects of data, i.e., aspects of data as "dedomena" and/or signals. That is to say, we are unconcerned about the physical medium (and corresponding signals) in which data is stored or transmitted, e.g., transistors (on/off states), switches (open/closed states), electric circuits (high/low voltages), discs or tapes (magnetized and non-magnetized regions), CDs (presence and absence of pits). Nonetheless, it is important to notice that information will bear some limitations (e.g., storage amount, transmission speed) due to its ultimate reliance on physical representation (Floridi, 2010). For instance, storage limitation gives rise to the informational concern of history tracking (discussed on chapter 6).

What we consider relevant to the information level are the *symbolic aspects* of data, i.e., the way we communicate certain information through symbols. For example, to convey the information "Mary is 5 ½ feet tall", one has to transmit relevant sequences of symbols, such as "Mary" and "5 ½", that refer to things in reality. When humans deal with information systems, symbolic data (as opposed to physical data) is the ultimate object of input and output operations. A well-formed sequence of symbols is what is actually called in the database literature "data". This is akin to the definition of data as "lacks of uniformity between two symbols". Accordingly, henceforth when we use the term "data" without further qualification, we mean symbolic data.

We could compare our usage of "data" with a definition provided by the Oxford English Dictionary (OED, 2009), which defines data as "the quantities, characters, or symbols on which operations are performed by computers and other automatic equipment, and which may be stored or transmitted in the form of electrical signals, records on magnetic tape or punched cards, etc.". This definition emphasizes the symbolic aspect of data ("quantities, characters, or symbols") and also mentions an operational aspect ("on which operations are performed" and "which may be stored or transmitted") as well as the underlying physical aspect ("in the form of electrical signals, records on magnetic tape or punched cards, etc."). Finally, our vision is also aligned with that of Langefors:

> It becomes immediately obvious that if data are what are processed by computers (or other means) then the information which people get from the data is something distinct from the data. If data may aid people in making decisions or performing actions, this must mean that the data inform people in the sense of making something known to them. It is clear then that the data must represent something that can also be expressed by natural language. Clearly then, in this sense, written sentences of natural language (as well as spoken or recorded sentences of any kind) are also data, that is, they are sets of signs representing knowledge or information. (Langefors, 1980)

3.2 INFORMATION

As accounted by (Langefors, 1980), data alone cannot "carry" information. In fact, a piece of data has to be properly interpreted by an informational agent so information can be extracted. In the following, we investigate important aspects of data interpretation.

According to Langefors, data, at best, gives rise to "information in the minds of people and only in those people who hold a suitable (...) world view (...) in their mind". This so-called "world view" receives a multitude of names in (Langefors, 1980) such as "user view", "frame-of-reference", "receiving structure", "semantic background", "general background knowledge", "infological model", "the model of the part of reality being involved" and "the view of the world held by the users of data". Here, we consider this "world view" to be equivalent to what we call a domain conceptualization. Langefors' statement highlights two important aspects of data interpretation. First, the segment "information in the minds of people" stresses that data interpretation acts over the symbol realm (data) and creates something in the thought realm (information). Second, data interpretation requires the adoption of a domain conceptualization. Furthermore, Langefors also indicates that data interpretation requires knowledge of data syntax (information structure) or, in his terms, "the data representation" or "the language of the intended user". Ultimately, we conclude that "the information that may be conveyed by a set of data, D, depends on the person receiving the data" or, in our terms, the informational agent performing the data interpretation.

As previously mentioned, well-formed and meaningful data results in semantic content. In line with (Floridi, 2010), we consider that there are two sorts of semantic content: *factual* and *instructional*. Instructional semantic content "is not about a situation, a fact, or a state of affairs *w* and does not model, or describe, or represent *w*" (Floridi, 2010). Examples include an invitation ("you are cordially invited to the college party"), an order ("close the window!") and an instruction ("to open the box turn the key"). At the information level, we restrict our characterization to information as *factual semantic content*, i.e., information that refers to phenomenon in reality. As explained by Langefors, understanding some data will imply "the imagining of the [phenomenon in reality] as the observer perceived it. This would mean that the user of data would *conceive* (...) the part of the world that was *perceived* by the observer who originated the message" (Langefors, 1980). At this point, it is worth to cite Langefors' example of data interpretation (we highlight the aspects of information as factual semantic content):

> Let us consider a specific data term, such as the number 17 or, more precisely, the printed sign "17". (...) Now, let us assume that we are told that the term "17" is a quantity on hand in a store. This attaches some (...) meaning to the term. Accordingly if the data term "17" is supplemented with the data term "Quantity-on-Hand, 17" some more precise specification of meaning is

> conveyed. But still we do not obtain information. Instead, if we are presented with the sentence "The quantity in store of articles of type A is 17 pieces" we feel having been informed. We might draw some conclusions from such a statement, provided **we perceive what it will look like in the stock room or how we might proceed if we would want to verify the sentence**. Thus it adds to our view of the world. It seems then that a text string, thus a group of data terms, in order to convey information not only must have meaning, it must have a truth value also - **it must state a fact and would be regarded as false if contradicted by real facts**. (Langefors, 1980)

In addition, we cite another Langefors' example to evidence that data by itself does not carry information, but rather there are premises to be taken in data interpretation. Let us assume that the following data (sentences) represent the same information:

(1) "The quantity in store of articles of type A is 17 pieces"

(2) "There remain 17 pieces of article type A for disposal"

(3) "Of article A there are 17 pieces on hand"

(4) "We are now left with 17 pieces"

(5) "Qty-on-Hand (A, 17)"

What can be noticed is that sentence (4) implicitly refers to "articles of type A", thus requires a particular context. Additionally, sentence (5) "calls for a particular syntax/semantics descriptions for each individual predicate" (Langefors, 1980). Ultimately, every data must make known: what entities it is intended to inform about (reference context), what it makes known about those entities (property context) and the time during which those entities hold the properties (time context). In data interpretation, the interpreter must be aware of those contexts in order to properly extract information. Moreover, as a consequence, the same piece of information may be structured in many ways in data, as long as contexts are established.

Information as factual semantic content could be related to what is called "proposition" in philosophy literature, as noticed by Langefors (in bold, we highlight aspects of information as semantic content):

> (...) in logic there is sometimes made a distinction between a *declarative sentence* (which is intended to state a fact) and the *proposition* which is formulated or represented by the sentence. The proposition is **"the cognitive content"** of the sentence and **"the thought of a fact"** which may or may not prevail. It is clear, now, that the proposition (in the logical sense of above) is the *information to be conveyed* by the sentence and the sentence is a group of data used to represent that information (the proposition). (Langefors, 1980)

In the philosophy literature, a property that is commonly ascribed to propositions is that of being *objects of belief*, as accounted by Moore:

> When we hear certain words spoken and understand their meaning, we may do three different things: we may *believe* the proposition which they express, we may *disbelieve* it, or we may simply *understand* what the words mean, without either believing or disbelieving it. (…) The difference between the three cases merely consists in the fact, that when we believe or disbelieve, we *also* do something else *beside* merely apprehending the proposition: beside merely apprehending it, we also have towards it one attitude which is called belief, or another different attitude which is called disbelief. (Moore, 1953)

Accordingly, we assume here that informational agents hold an attitude of belief towards information. To summarize the discussion on data interpretation and information as factual semantic content, we illustrate in Figure 3.1 how information is communicated by means of data.

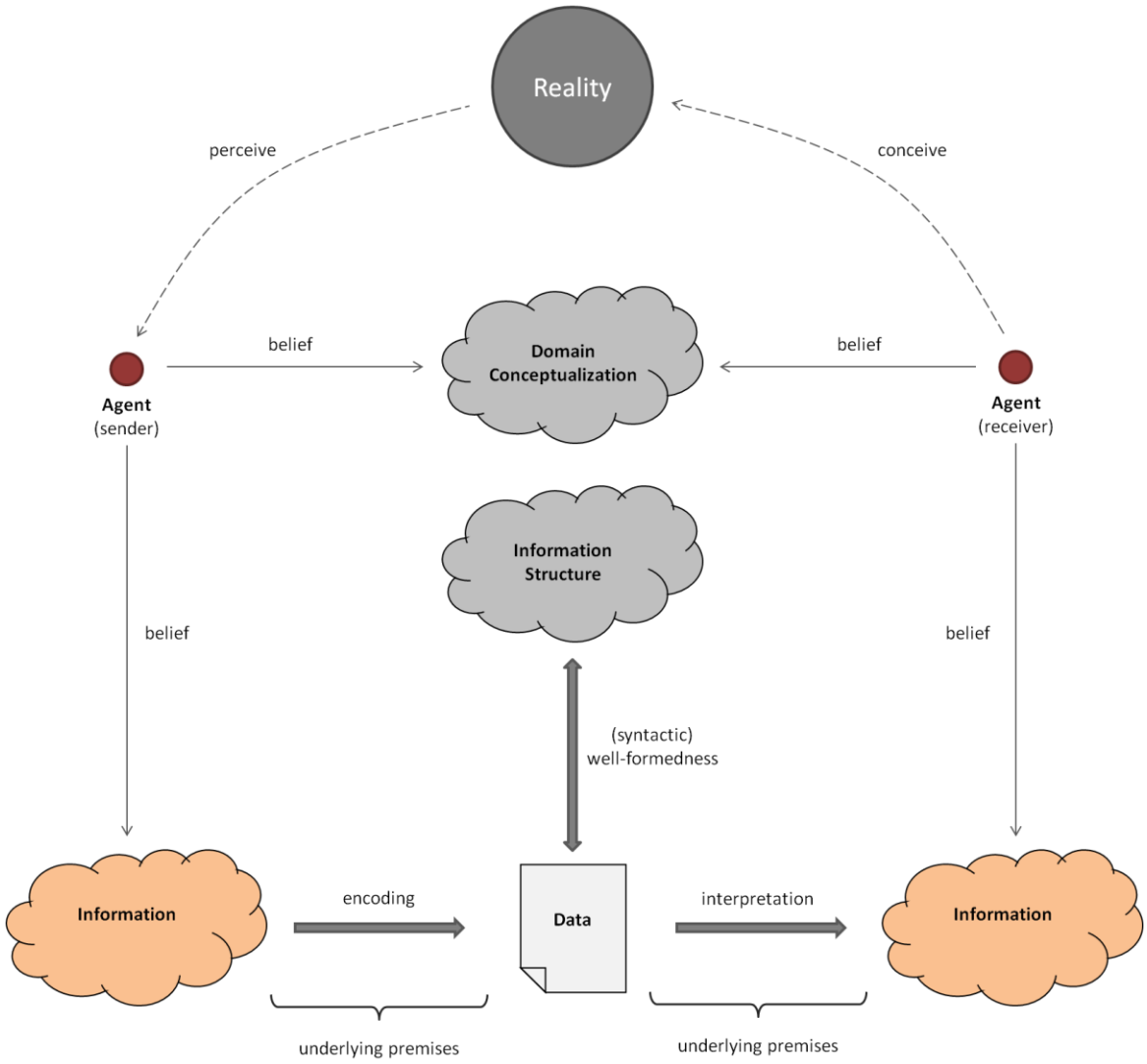

Figure 3.1 - Information being communicated by means of data

Suppose one informational agent (the sender) has a piece of information about certain phenomenon in reality and wishes to communicate it so the other informational agent (the receiver) can be aware

of the phenomenon. Furthermore, suppose this piece of information is in conformity with the domain conceptualization that is adopted by the sender. In order to do such, the sender must *encode* information, which is inside his mind, in data, which in turn must conform to syntactical rules provided by an information structure. In addition, the resulting piece of (symbolic) data must be *embodied* in physical data. Afterwards, the piece of (symbolic) data must be *transmitted* by the sender and *obtained* by the receiver in a process that relies on a physical medium. Once the piece of data has been properly obtained, the receiver must perform an *interpretation*. We assume the receiver adopts the same domain conceptualization and information structure that were adopted by the sender. Once information has been extracted from data, the receiver *conceives* the possible phenomenon in reality and may hold an attitude of *belief* towards it, thus taking the information to be true.

3.3 Informational Concerns

We summarize the aforementioned aspects of information and data by means of the triangle of reference depicted in Figure 3.2. As we explain the figure, we characterize two sorts of informational concerns, namely, *information demand concerns* and *representation concerns*.

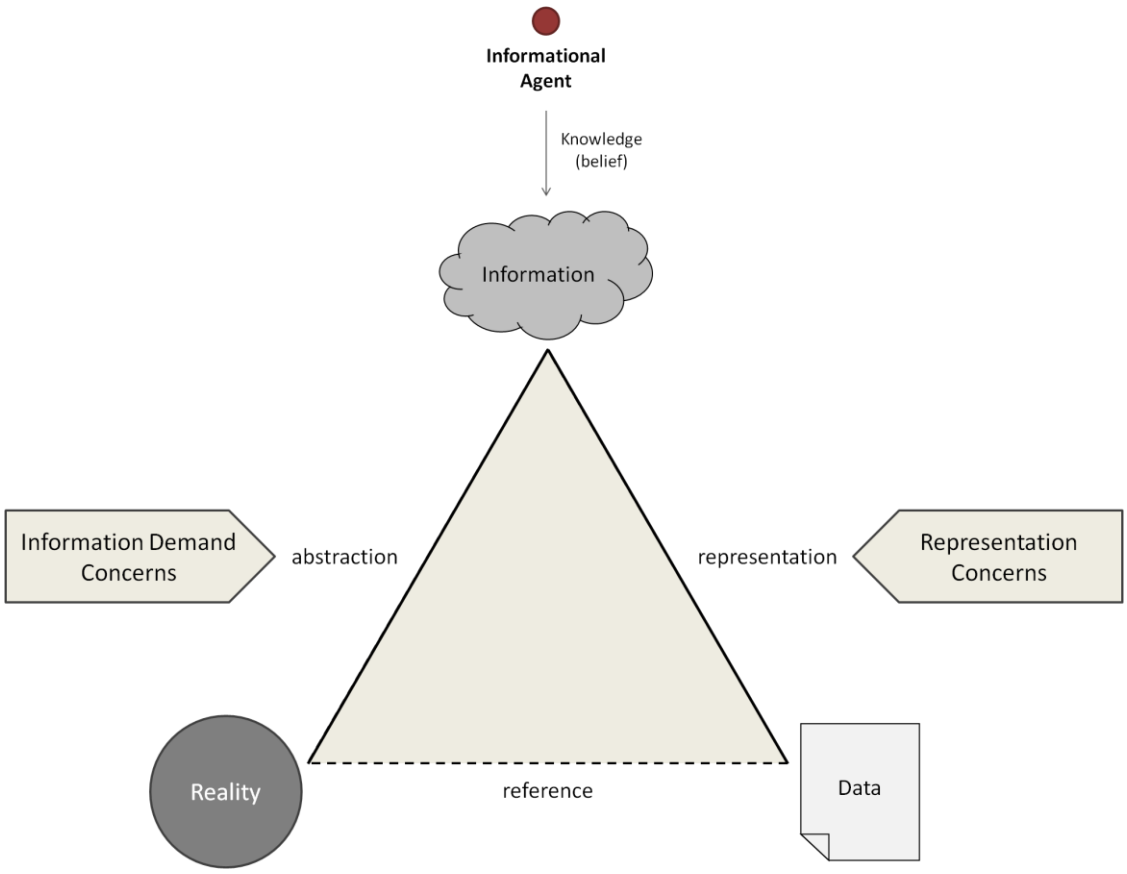

Figure 3.2 - The triangle of reference for information and data

3.3.1 Information Demand Concerns

Information belongs to the thought realm and is the semantic content of beliefs of informational agents. The left edge of the triangle, named "abstraction", illustrates the relation between information and reality. Information attempts to describe phenomena in reality, but may fail to do so or may do it in an incomplete manner, given the domain conceptualization adopted by the informational agent. There are several reasons why information may not accurately describe reality w.r.t. a domain conceptualization.

Foremost, there are cognitive limitations in the process of perceiving reality (in terms of a domain conceptualization) and creating information. Some information require measurement, i.e., the use of an instrument (e.g., "Mary is 5 ½ feet tall"); other information simply require the basic senses, without instrument assistance (e.g., "Mary is a woman"). Either way, the act of grasping reality is susceptible to flaws, thus information can potentially be false, inaccurate (close to the truth, but not the truth), incomplete and unknown. Secondly, information manipulation (storage, transmission, processing, etc.) involves symbolic data and ultimately physical data. Thus, the capacity to hold and manipulate information is not unlimited, and information may be *stored* when deemed relevant and *discarded* when deemed irrelevant. Finally, informational agents may have strategic goals that are fulfilled by partial descriptions (in terms of a domain conceptualization).

Thus, the "abstraction" edge in the triangle represents issues on how well information describes phenomena in reality w.r.t. a domain conceptualization. Generally speaking, the relation between information and reality is marked by *information demand concerns*, which determine what the informational agent is required to know about phenomena of interest (taking into account cognitive and physical limitations, as well as strategic goals)[5].

In this thesis, we identify the following information demand concerns: scope, history tracking, time tracking and measurement. Information demand concerns involve the selection of which types of information are relevant to be known (and ultimately stored) by an informational agent. Those concerns are further categorized in terms of the metaphysical aspects they focus on: (i) history tracking is based on the distinction between current and past information, (ii) time tracking is focused on information about timing aspects of things, (iii) measurement is focused on information about properties of things, and (iv) scope is centered on information about certain categories of being (of a domain conceptualization).

As a final note, issues on cognitive limitations are commonly attributed to the branch of philosophy called *epistemology*, which studies the nature and sources of knowledge. For this reason,

[5] What we call here "information demand" could also be called "information requirement(s)" or "information needs".

we could use the term *epistemological concerns* to refer to those issues. According to our discussion, the "abstraction" edge may be (in part) marked by those epistemological concerns. Nevertheless, this is not to say that the ontological level is completely free from epistemological concerns. As a matter of fact, a domain conceptualization involves cognitive categories that only exist because cognitive agents exist. This view is defended by the so-called constructivist epistemology, which argues that we *propose* concepts in order to explain our *sensory experience*, based on the premise that *we cannot objectively know reality*. Hence, epistemological concerns are somehow present in both the ontological and the information levels. Our point is that, given a domain conceptualization (which is already affected by some epistemological concerns), an agent's information demand *may be* affected by additional epistemological concerns.

3.3.2 Representation Concerns

The right edge of the triangle, called "representation", illustrates the relation between information and data. Data belongs to the symbol realm and materializes information so it can be further stored, communicated and processed. Data by itself does not completely carry information. Rather, an informational agent must extract information from data by means of an interpretation process, which requires knowledge of the domain conceptualization, the information structure (data syntax) and underlying contexts (reference, time-space, etc.). In other words, data only represents phenomena in reality by means of information (factual semantic content). For this reason, the relation between data and reality is indirect and is illustrated by the dotted edge in the triangle, called "reference". The relation of reference is as follows: a piece of data is supposed to refer to phenomena in reality, while data fragments within it are supposed to refer to, e.g., things in reality, properties of things, relations between things. As a consequence, the relation between information and data is marked by *representation concerns*, which determine how information should be encoded in data (taking into account data syntax and reference).

In this thesis, we identify the following representation concerns: the lexicalization of reference schemes and data types, and the selection of the information modeling technique. Lexicalization of reference schemes is focused on the representation of data types that stand for individuals in reality, the so-called identifiers. Lexicalization of data types concentrates on the representation of data types that stand for measured properties of individuals in reality. Information modeling techniques contemplate how symbolic data is fundamentally organized (e.g., object-oriented, entity-relationship). We extend the discussion on this subject in the following section.

3.4 Information Modeling

Ultimately, information modeling is about providing the syntax of well-formed data[6]. At first glance, this would only involve what we called representation concerns. Nonetheless, data is supposed to represent information that, in turn, stands for phenomena in reality. As a result, a specification of data syntax implicitly settles what sorts of information could be known by the users of data. Ergo, the role of information modeling is two-fold, addressing both representation and information demand concerns, together forming what we call *informational concerns*.

Among the identified informational concerns, we should make a special commitment w.r.t. the selection of an information modeling technique. More specifically, we select one particular technique to be used throughout this thesis. That is to say, we assume that the addressing of this informational concern is the same for all informational agents. The selection of an information modeling technique is not a trivial task, as accounted by Simsion and Witt:

> One of the challenges of writing a book on data modeling is to decide which of the published data modeling "languages" and associated conventions to use, in particular for diagrammatic representation of conceptual models. There are many options and continued debate about their relative merits. Indeed, much of the academic literature on data modeling is devoted to exploring different languages and conventions and proposing DBMS architectures to support them. (Simsion & Witt, 2005)

To a great extent, the selection of an information modeling language relies on pragmatic and non-technical issues. For example, based on pragmatic criteria, Simsion and Witt narrowed their selection to two options, namely, the Entity-Relationship (ER) approach and the Unified Modeling Language (UML). According to them, "the overwhelming majority of practicing modelers know and use one or both of these languages" and "tools to support data modeling almost invariably use E-R or UML conventions" (Simsion & Witt, 2005). Here, we select the UML language which "provides conventions for recording a wide range of conventional and object-oriented analysis and design deliverables, including data models represented by class diagrams" (Simsion & Witt, 2005).

It is worth clarifying the role of what we call an information model. In the data modeling literature, there is a common distinction between conceptual and logical data models. On the one hand, the conceptual data model is "a (relatively)[7] technology-independent specification of the data

[6] As we have discussed in Chapter 1, this means that "information modeling" is actually a misnomer, and "conceptual data modeling" would be more appropriate here. Henceforth, we use the terms interchangeably.

[7] Simsion and Witt use the term "relatively" as most information modeling languages were designed in a bottom-up manner, partially taking into account implementation details. For example, UML is biased towards software design and ER is biased towards database technologies.

to be held in the database" (Simsion & Witt, 2005). Typically, a conceptual data model would be written in the Entity-Relationship approach. On the other hand, the logical data model is "a translation of the conceptual model into structures that can be implemented using a database management system (DBMS)"; "today, that usually means that this model specifies tables and columns" (Simsion & Witt, 2005). Typically, a logical data model would be committed to a specific database approach such as relational (tables and columns), network or hierarchical.

Accordingly, what we call an information model refers to something conceptual (as opposed to logical), i.e., a *relatively* technology-independent specification of data. That is to say, an information model is not focused on any database or programming language implementation. As a consequence, it is a specification about abstract data types; not about database tables or programming language classes. Nonetheless, we cannot deny the role of an information model as a specification to be further used in the design and implementation of information systems. Therefore, during information modeling, we advance some almost inevitable implementation commitments that are commonly addressed during a further implementation phase. We do such in order to bridge the gap between a domain ontology and an implementation (or logical) model.

Although we consider that information modeling deals with abstract data types, there are still multiple ways to structure data, given the same information demand. For example, one possible way to structure abstract data is by means of object orientation, i.e., pieces of data are represented as objects that instantiate classes (data types). Further, when object orientation is chosen, there are still alternatives concerning, e.g., static versus dynamic classification, single versus multiple classification (Fowler, 2003). Thus, we inevitably have to commit to certain *design decisions* in order to build information models. Those decisions could be parameters of our model-driven approach, not for addressing differences in information demand, but rather differences in design. Nevertheless, as a matter of scope, we focus our work exclusively on decisions about information demand (viz. informational decisions). We do not attempt to discuss what design decisions are the most suitable, but rather we adopt decisions that are usually well-accepted in the information and data modeling literature.

Our choice for UML class diagrams implies that we adopt an object-oriented approach for information modeling (a design decision). UML class diagrams have no intrinsic limitations with respect to features such as dynamic and multiple classification. Nonetheless, those features are commonly avoided in object-oriented approaches, as their absence in information models facilitates further object-oriented, as well as relational, implementations. This motivates our design decision to avoid dynamic classification in the information models presented here. Those issues are further addressed in chapter 4.

3.5 INFORMATIONAL DECISIONS

Hitherto, we considered that the addressing of informational concerns is an unavoidable step for information manipulation. Since we presented those concerns as being exclusively related to information manipulation, then, by definition, they fall outside of the scope of the ontological level. Having settled a domain conceptualization at the ontological level, we assume that each informational agent may have particular ways of addressing each informational concern, due to differences in information demand. Hence, the addressing of informational concerns requires what we call *informational decisions*. In our model-driven approach, we use a domain ontology as a starting point for the construction of an information model, in a process that is guided by those informational decisions.

In the remainder of this chapter, we illustrate a design trajectory for three different information models based on the same domain ontology. This shows the variety of informational decisions that will be addressed in the remaining chapters. We consider that each information model fulfills the information demand of a certain informational agent. This is depicted in Figure 3.3, where we present a human agent (HA), an information system (IS) and an artificial intelligence agent (AA).

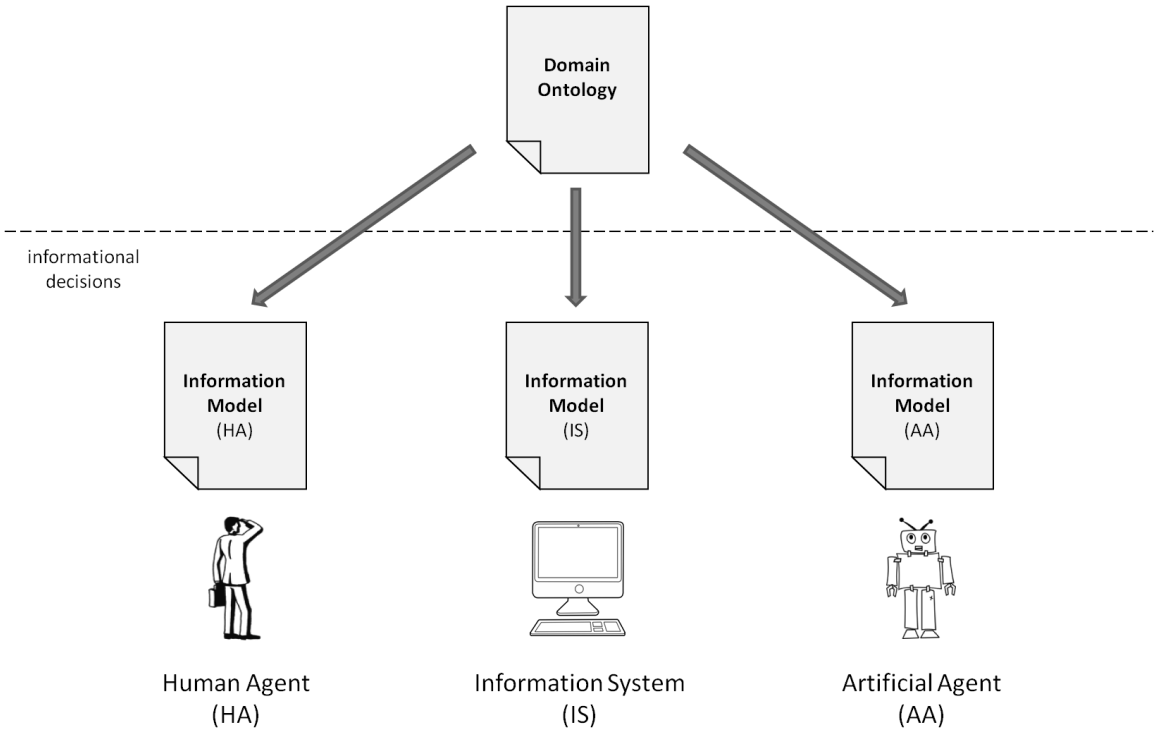

Figure 3.3 - Several information models according to different informational decisions

For our illustration, we consider the domain ontology about employments depicted in Figure 3.4, which is a fragment of the running example of Figure 2.8.

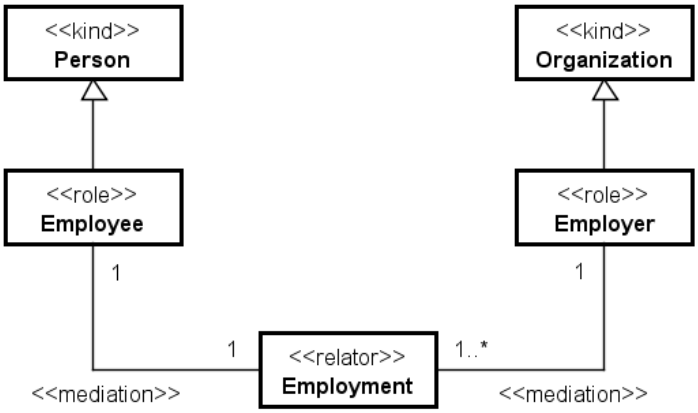

Figure 3.4 - A fragment domain ontology shared by the informational agents

First of all, we consider that the human agent (HA) has a full information demand, which is addressed by the information model of Figure 3.5. With respect to scope, the agent must be able to store information about entities that instantiate any of the universals specified in the domain ontology, without restrictions. This is achieved by means of the types (classes) Person, Organization and Employment. Types are connected via relations (associations) to express data on role playing.

With respect to history tracking, the agent must store data on past and present entities, and must be able to distinguish whether some data is about a present entity or not. Thus, each type has a boolean attribute named “current”. With respect to time tracking, the agent must be able to store information about all timing aspects of entities, viz. start and end time of existence, as well as duration. This is achieved by the attributes named “start”, “end” and “duration”, respectively. In terms of reference, the agent identifies entities by means of integer identifiers, namely, the “id” attribute of types. For measurement, data types for time instants and time durations have to be specified (for the sake of simplicity, they have been omitted in the information model).

We present some sample data that conforms to this information model in Figure 3.6. Each circle represents a data fragment (object) that instantiates the type (class) of the corresponding column (Person, Employment or Organization). Lines represent relations between data fragments. Consequently, our sample contains data about two people, three employments and two organizations. Moreover, each data fragment contains instances of class attributes (viz. id, current, start, end and duration); several data formats are shown for time tracking attributes.

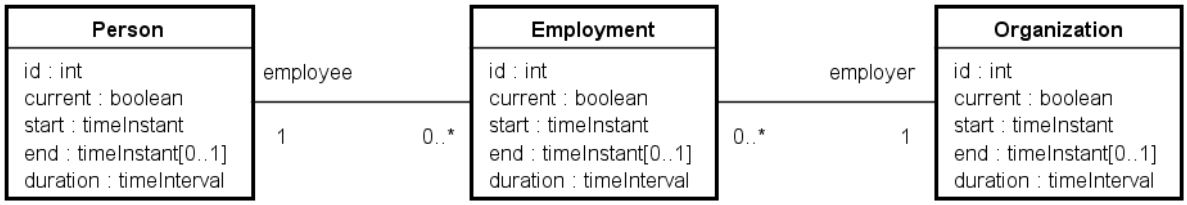

Figure 3.5 - Information model (HA)

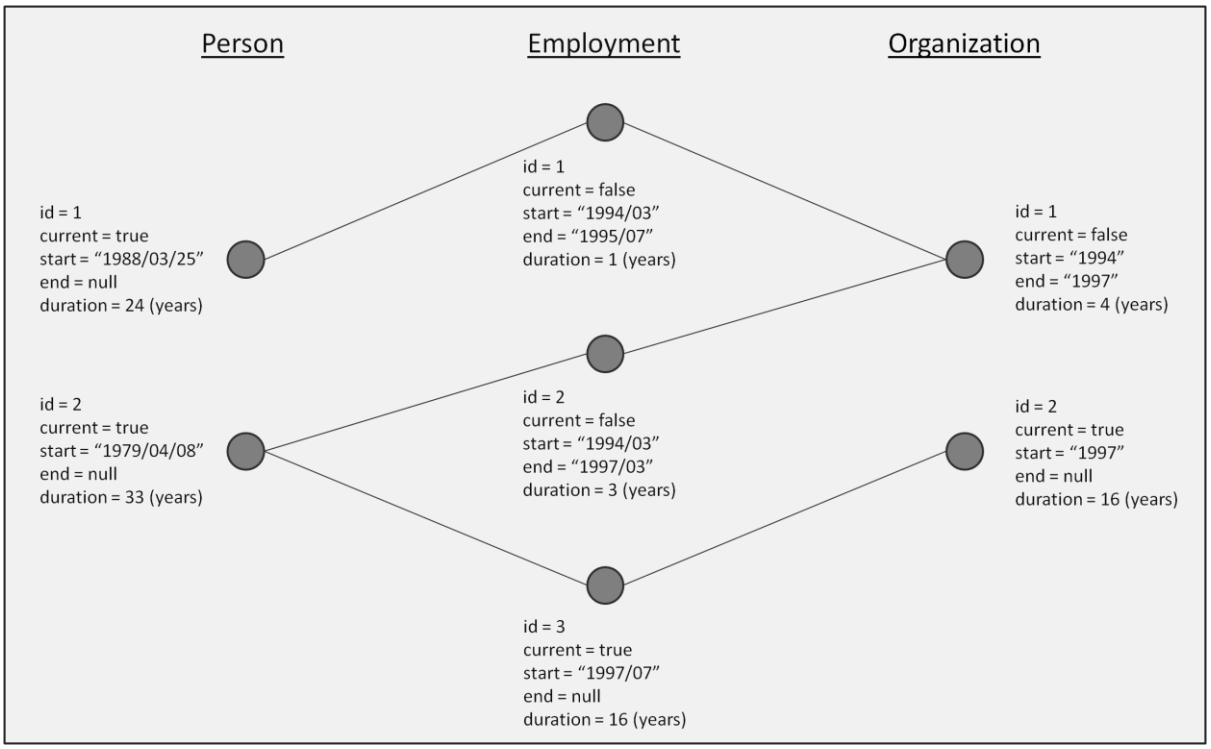

Figure 3.6 - Sample data (HA)

Suppose the information system (IS) has a partial information demand, addressed in the information model of Figure 3.7. With respect to scope, only information about entities that instantiate the Organization and the Employment universals is required. With respect to history tracking, the information system only stores information about current (active) organizations and about current (active) employments. With respect to time tracking, the agent only stores information on the start time of employments (encoded in a year granularity). In terms of reference, suppose the information system only stores information about Brazilian organizations and thus identify them by a number issued by the Brazilian government, namely, *Cadastro Nacional da Pessoa Jurídica* (CNPJ). In addition, suppose no identifier is required for employments (assuming that objects have an intrinsic identity in object-oriented approaches). As an illustration, sample data conforming to this information model is depicted in Figure 3.8.

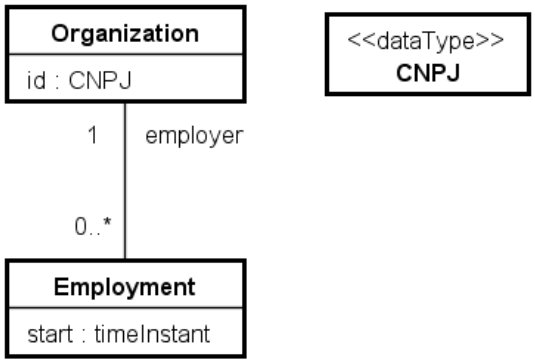

Figure 3.7 - Information model (IS)

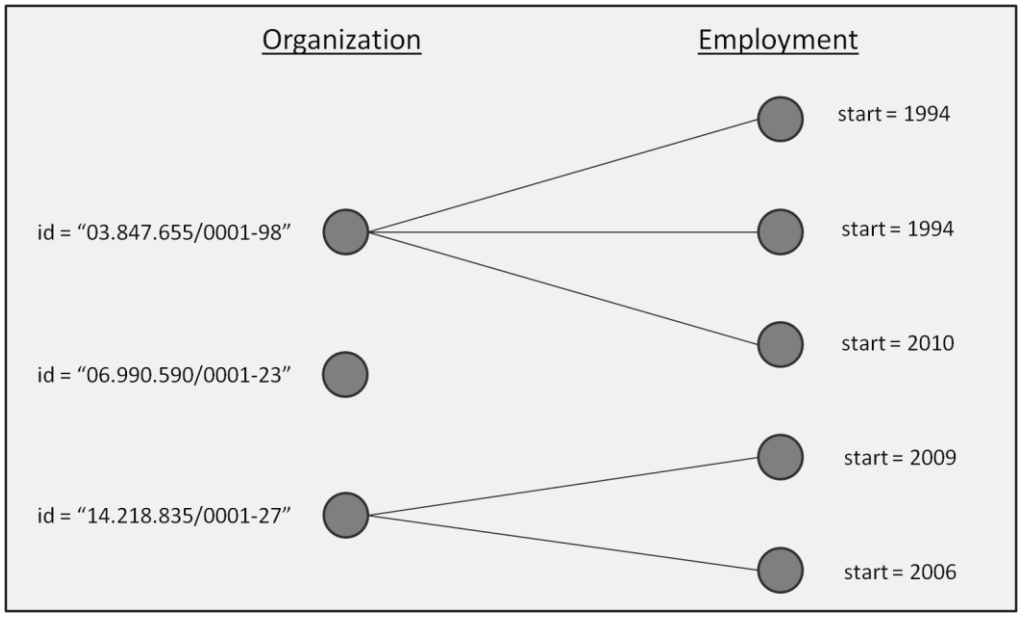

Figure 3.8 - Sample data (IS)

Finally, suppose the artificial intelligence agent (AA) also has a partial information demand, addressed in the information model of Figure 3.9. With respect to scope, suppose the agent stores information about a single organization; thus, only information about people and employments is required. With respect to history tracking, suppose the agent must know about current (alive) people and only about their past employments. For time tracking, the agent is only required to know about the duration of (past) employments (encoded in years). For reference, suppose the agent stores information about an American organization and refers to people in terms of Social Security Number (SSN). We exemplify sample data conforming to this information model in Figure 3.10.

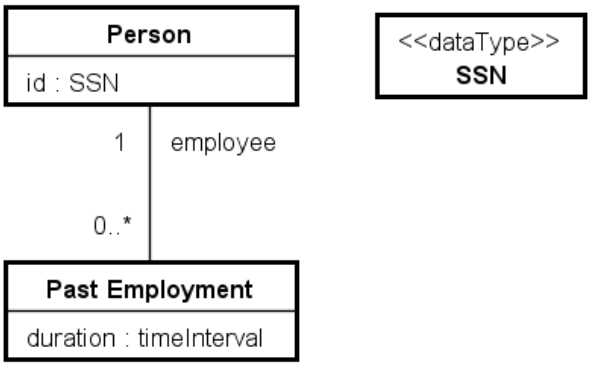

Figure 3.9 - Information model (AA)

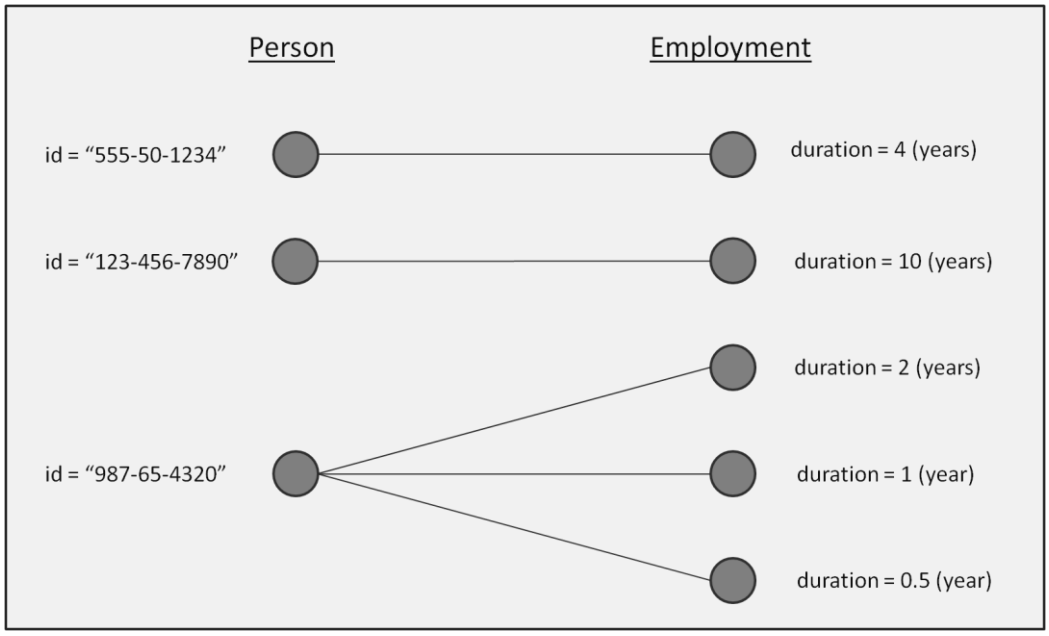

Figure 3.10 - Sample data (AA)

At last, Figure 3.11 illustrates the several informational decisions taken over the same domain ontology to reach the provided information models for each informational agent.

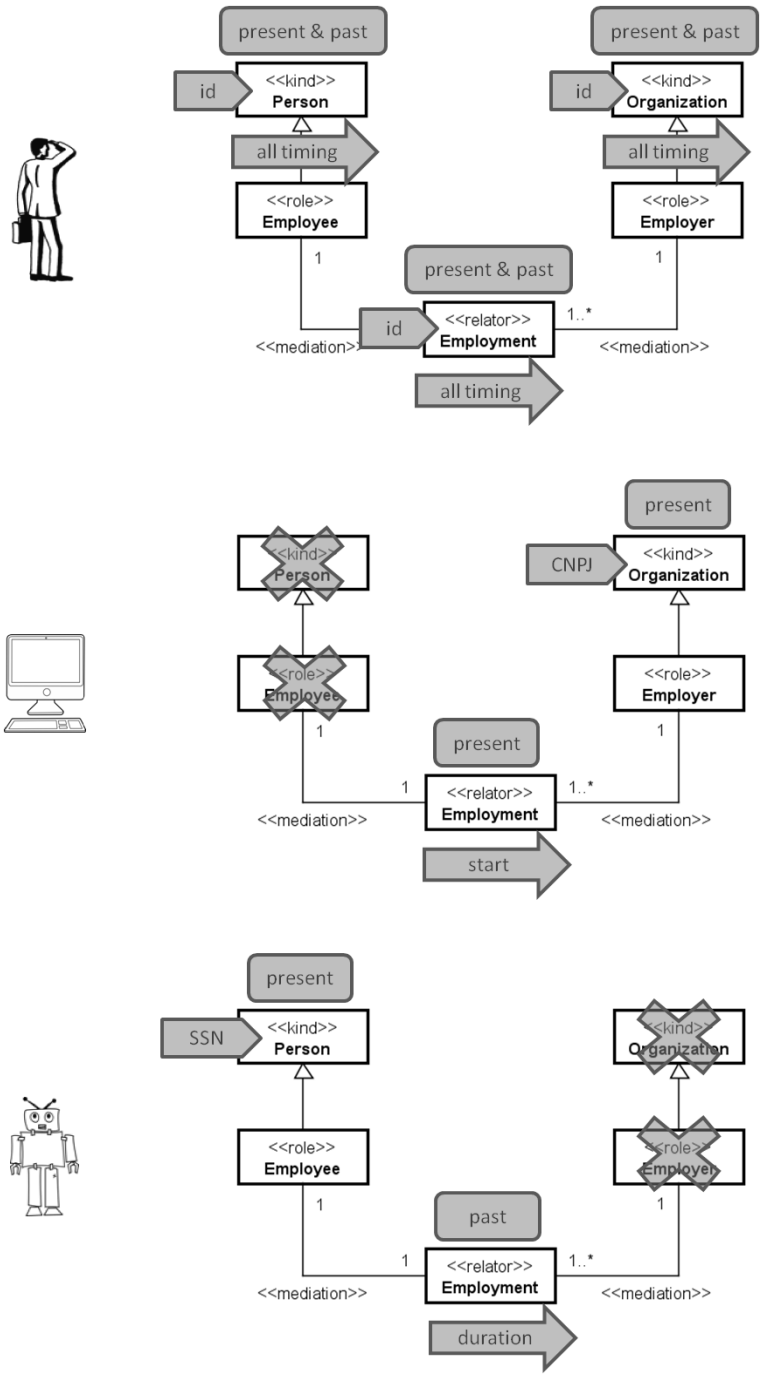

Figure 3.11 - Informational decisions over a domain ontology

3.6 CONCLUSIONS

In this chapter, we characterized “data” as symbolic data and contrasted this notion with the one of physical data, which we disregard at the information level. Afterwards, we discussed the process of data interpretation, which gives rise to information in the minds of informational agents. We concluded that, in order to properly interpret a piece of data, an agent must adopt a domain conceptualization to which the corresponding piece of information conforms. We also assumed that data is structured according to a syntax that is known by the agent. We explained that data interpretation involves several underlying premises that are not contained in data itself, but rather in the minds of agents using data.

We characterized “information” as factual semantic content, which intends to refer to phenomena in reality. At the information level, we disregarded other sorts of semantic content (e.g., instructional) that do not possess such characteristic. In general, our view of “information” is related to that of “proposition” in philosophy. Akin to what is commonly ascribed to propositions, we assumed that informational agents hold an attitude of belief towards information.

After that, we characterized two sorts of informational concerns, namely, information demand concerns (related to information and reality) and representation concerns (related to information and data). We also discussed the role of what we called epistemological concerns in both the information level and the ontological level.

Then, we described the information modeling technique adopted in this thesis. In our approach, a resulting information model is a relatively technology-independent specification of data, which bridges the gap between a domain ontology and an implementation model. We explained that our approach inevitably commits to some design decisions, as there are multiple ways to structure data, given the same information demand. For instance, we chose UML class diagrams for specifying information models and we chose to avoid dynamic classification. As we stated, we focus our model-driven approach on informational decisions, instead of design decisions.

Finally, in the remainder of the chapter, we illustrated the core approach that should be elaborated in the following chapters.

4 From a Domain Ontology to an Object-Oriented Information Model

As we have discussed in chapter 2, a domain ontology specifies a number of universals used to classify entities in reality. In contrast, as we have seen in chapter 3, an information model specifies a number of types (classes) used to classify data fragments (objects), which carry information about entities in reality. This chapter addresses the first challenge in our model-driven approach, namely, specifying the structure of types of an information model, based on the structure of universals of a domain ontology.

We take into account that the OntoUML language relies on the dynamic classification of entities in reality, while UML class diagrams used at the information level will refrain from using dynamic classification of objects (data fragments) in order to simplify further implementation efforts. As previously discussed, avoiding dynamic classification in UML information models is a particular design decision that we commit to in this thesis (and not a limitation of UML).

At the information level, the subject of dynamic classification is strongly related to storing data about role playing of entities. The literature on information modeling and object-oriented modeling proposes several patterns for representing the dynamics of role playing without the use of dynamic classification. Nevertheless, they disregard the relational dependency of roles and, consequently, the proposed patterns are not completely suitable for our purposes. Therefore, we analyze the proposed patterns for representing dynamic aspects in information modeling and then we propose a modified approach.

For explanation purposes, we restrict this chapter to cases of full information demand w.r.t. history tracking and scope. The domain ontology fragments to be further presented here belong to the running example depicted in Figure 2.8. We initially explain the straightforward transformation pattern for static aspects and, afterwards, we discuss the addressing of dynamic aspects.

4.1 Static Aspects: Addressing Kinds, SubKinds and Categories

At the ontological level, substantials may be rigidly classified into some categories of being (universals). Accordingly, at the information level, we assume that an object (data fragment) carrying information about static aspects of a substantial is statically classified into types. In our model-driven approach, rigid universals in the domain ontology provide a base for creating types in the information model. We discuss about those types in the following.

At the ontological level, every substantial along its whole lifecycle must instantiate exactly one and the same Substance Sortal that provides a principle of identity for it (e.g., Person, Organization).

In addition, every substantial, along its whole lifecycle, can be further categorized in terms of other Rigid Sortals besides the ultimate Substance Sortal, viz. SubKinds (e.g., Man, Woman). Because Kinds and SubKinds are also known as “natural kinds” (Bunge, 1977), we call the types corresponding to them at the information level “natural types”.

Figure 4.1 depicts, in the upper part, a domain ontology (fragment) specifying Kinds and SubKinds and, in the lower part, a corresponding information model (fragment) specifying natural types. Arrows in the figure depict the relation between universals in the domain ontology and types in the information model. Henceforth, for convention, when we refer to a certain type (e.g., Organization type) we assume it corresponds to the universal bearing the same name (e.g., Organization Kind).

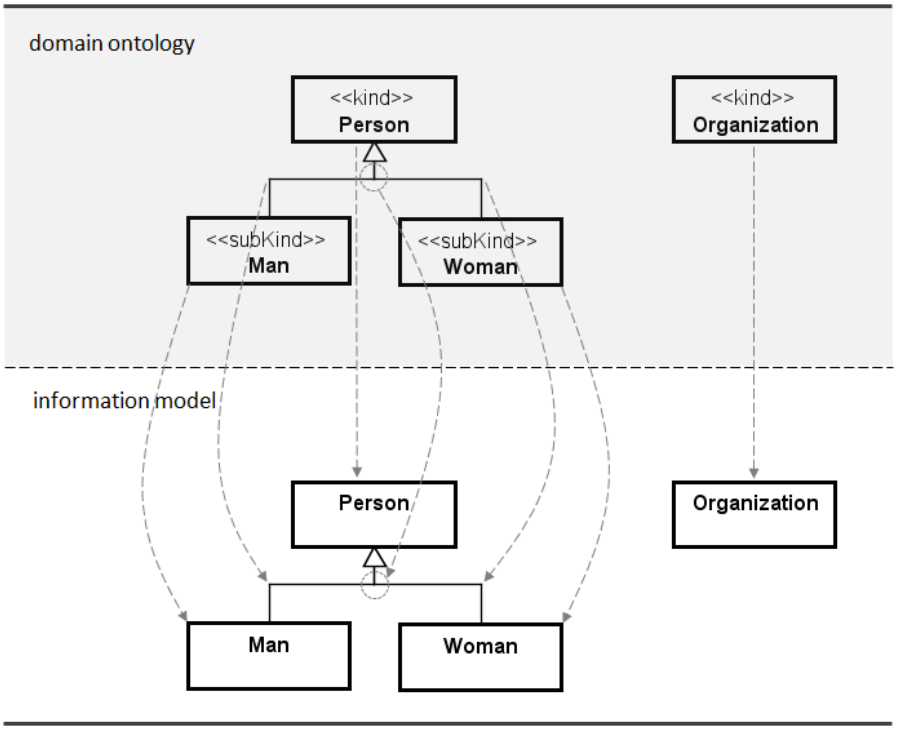

Figure 4.1 - Our information modeling pattern for Kinds and SubKinds (via natural types)

A generalization relation between two Rigid Sortals in a domain ontology is reflected in a generalization relation between the corresponding natural types in the information model. For example, due to the generalization relation between Person and Woman in the domain ontology, a generalization between the Person type and the Woman type is created in the information model.

A generalization set involving generalizations between Rigid Sortals is reflected in a generalization set involving the corresponding generalizations between natural types. For instance, the Man type and the Woman type imply two generalizations, namely, one between the Man type and the Person type and other between the Woman type and the Person type. Those generalizations, in turn, imply a generalization set involving them.

In addition, at the ontological level, a substantial throughout its lifecycle may be classified according to rigid Mixins (Categories). Correspondingly, at the information level, we assume that the object (carrying data on statics aspects of a substantial) may be statically classified into types corresponding to Categories. The pattern for Categories is straightforward and is depicted in Figure 4.2, which illustrates that the Legal Entity Category in the domain ontology (upper part) generates a corresponding Legal Entity type in the information model (lower part). Furthermore, a type corresponding to a Category generalizes other types in the information model, based on the generalizations of the corresponding universals in the domain ontology. For instance, the Legal Entity Category generalizes the Person Kind and the Organization Kind, therefore the Legal Entity type generalizes the Person type and the Organization type.

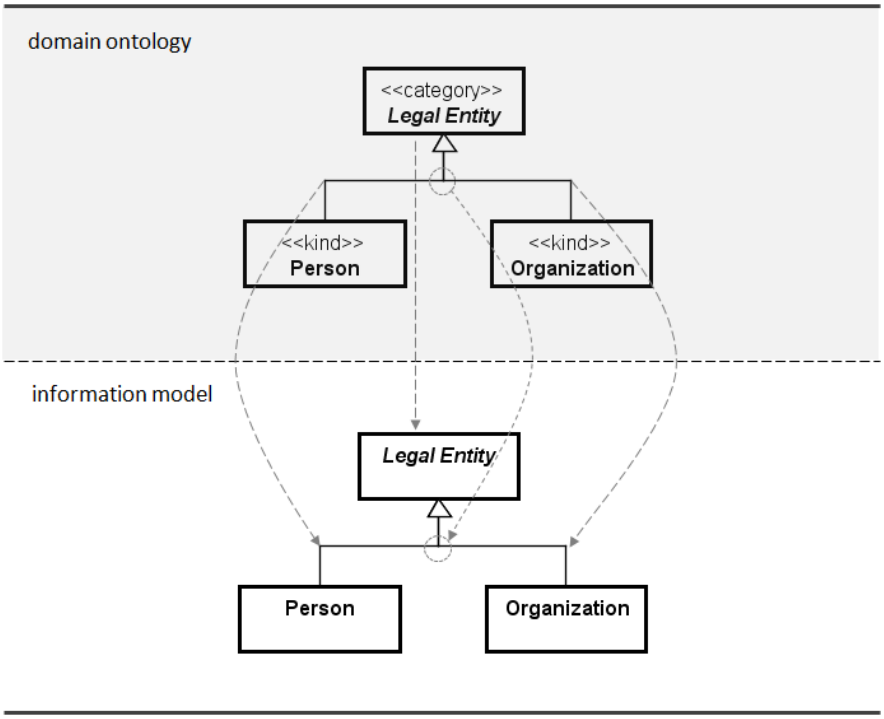

Figure 4.2 - Our information modeling pattern for Categories

4.2 Dynamic Aspects: Addressing Roles, Role Mixins and Relators

At the ontological level, every substantial may play various roles in the context of material relations (e.g., the substantial John plays the role of husband in his marriage with Mary and the same substantial John plays the role of employee of Apple Inc.). Our goal in this section is to provide an approach, at the information level, for the representation of data on dynamic aspects of things, more specifically, on role playing. In depth, we aim to convert structures in the domain ontology that involve Roles and Role Mixins, which rely on dynamic classification, into structures in the information model that do not rely on such classification mechanism.

Although several alternatives for the representation of roles in information modeling and in "object-oriented modeling" have been provided, none of those can be directly applied (at least in a suitable manner) in the context of our model-driven approach. This is due to our usage of a domain ontology as a starting point and due to the OntoUML representation of roles, which considers them in the context of material relations, involving Roles and Relators. As a consequence, we propose an approach that is a combination of two previously known approaches.

In order to characterize and justify our proposed approach, we present an in-depth discussion on background approaches. During the discussion, we take into account some important characteristics of roles, which have been summarized in (Steimann, 2000) and adapted here:

- Roles depend on relationships;
- A substantial may play different roles simultaneously;
- A substantial may play the same role (universal) several times, simultaneously. The main reason to distinguish multiple occurrences of the same role (universal) is that each qua individual is associated with a different state (e.g., employee's salary);
- A substantial may acquire and abandon roles dynamically;
- Substantials of different Kinds can play the same role;

After the background discussion, we present our approach, which is divided in the addressing of Roles and the addressing of Role Mixins.[8]

4.2.1 BACKGROUND

There are basically three kinds of representations of roles in object-oriented modeling (Steimann, 2000), namely, (i) roles as "named places" of relationships, (ii) roles as specializations and/or generalizations, and (iii) roles as "adjunct instances". In the following, we analyze those representations, discussing their advantages and disadvantages.

When roles are represented as named places of relationships, there is no such thing as role types. Instead, relationships are established between natural types (e.g., Person and Organization). Then, each "place" of a relationship bears the name of the role played by the corresponding natural type in the context of the relationship. In the UML class diagram, relationships are represented as UML associations and the "named places" referred to by Steimann are represented as role names of association ends. An example is depicted in Figure 4.3. In this information model, the employment relationship has two association ends, named "employee" and "employer", attached to the natural types Person and Organization, respectively.

[8] There are circumstances where roles may play roles, e.g., an Employee may be a Project Manager. We consider those situations to be outside the scope of this thesis, as those issues are not fully detailed in UFO.

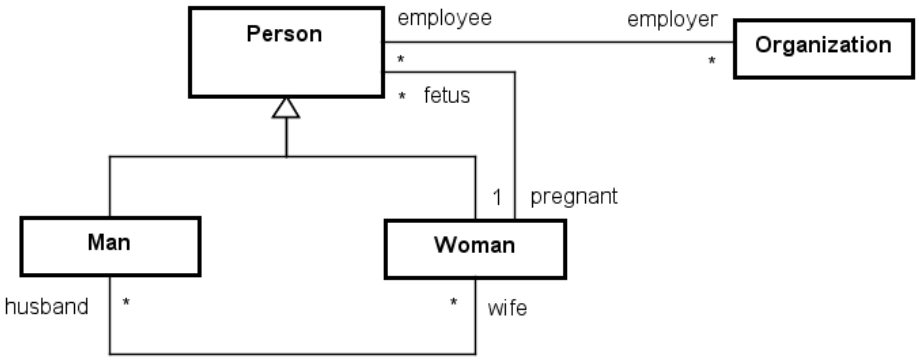

Figure 4.3 - Roles as named places of relationships

The advantage of this approach is that it “stresses that roles exist only in context” (Steimann, 2000), since role names only appear in the presence of relationships between natural types. Nonetheless, this approach “fails to account for the fact that roles come with their own properties (...), a deficiency that is usually resolved by regarding roles as types in their own right (and not as mere labels of types)” (Steimann, 2000).

This leads to the second approach, namely, roles as specializations and/or generalizations, which represents roles as types. As observed by Steimann, a role type is more specific than the natural type of role players, which would make it a specialization (and hence a subtype). An example of roles represented as types that specialize natural types is depicted in Figure 4.4.

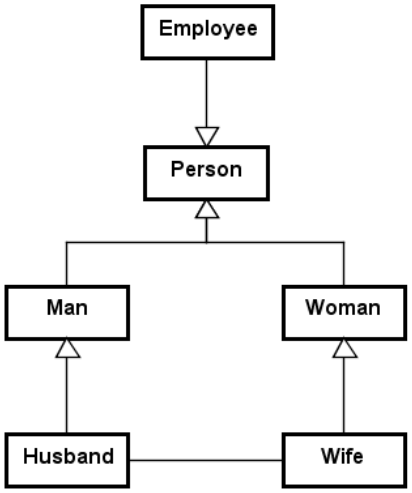

Figure 4.4 - Roles as specializations and/or generalizations

The problem is that such solution requires dynamic and multiple classification, since an object can change its roles and play several roles simultaneously (instance or object migration) (Steimann, 2000). This is not supported by most object-oriented approaches:

> Class-based object-oriented systems (...) represent real-world entities[9] as instances of the most specific class in which they can be classified. The association between an instance and a class is exclusive and permanent. Therefore, this approach is appropriate only if the real-world entities to be modeled[10] can be partitioned into a set of disjoint classes and never change their class.
>
> (Gottlob, Schrefl, & Röck, 1996)

Thus, object-oriented systems have serious difficulties in representing data about substantials (real-world entities) taking on different roles over time, if roles are represented as types (classes). An involved solution to these difficulties is discussed in (Gottlob et al., 1996), proposing that objects (data fragments) must be reclassified any time they evolve, in the following manner:

- An instance of the new class (representing the entity in the evolved state) must be created;
- Relevant information from the instance representing the old state of the entity must be copied to the new instance;
- All references to the old instance must be reset to the new instance;
- The obsolete instance must be deleted;

Moreover, further problems arise if an entity can take on several roles independently. In order to support exclusive classification of objects, a separate class must be defined for every possible combination of roles. These intersection classes are usually defined by means of multiple inheritance. Even further, class hierarchies cannot handle multiple occurrences of one entity in the same role type (e.g., a person being an employee in two organizations at the same time). (Gottlob et al., 1996)

This leads to the third approach, namely, roles as adjunct instances. An example is depicted in Figure 4.5. In this approach, roles are represented as types that are not specializations of natural types, and whose instances are carriers of role-specific state. At any point in time, an entity is represented by an instance of its natural type and instances of role types (representing the roles it currently plays). If an entity acquires a new role, a role-specific instance of the appropriate role type is created; if it abandons a role, the role-specific instance is destroyed. That is to say, a real-world entity is represented by several objects, each representing it in a particular role, e.g., somebody being a student and employed will be represented by a person object, a student object, and an employee object. (Gottlob et al., 1996)

---

[9] As we have advocated in this thesis, this statement would be better rephrased as “represent [data about] real-world entities”.

[10] “[data about] real-world entities to be modeled”

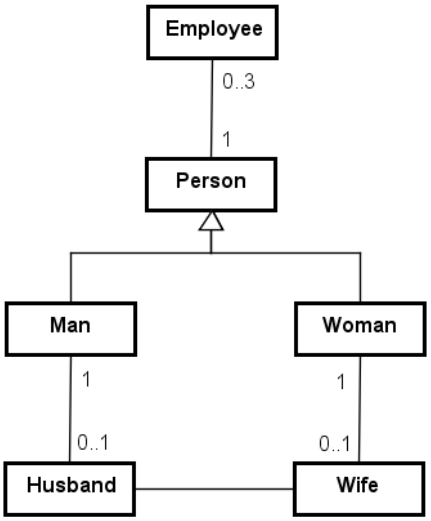

Figure 4.5 - Roles as adjunct instances

Using this approach, an object that instantiates a natural type may be related to several other objects that instantiate the same role type. For instance, a person object may be related to two employee objects. For those cases, one must provide ways to distinguish instances of the same role type, which motivates the creation of role identifiers (Wieringa, de Jonge, & Spruit, 1995).

Another advantage of roles as adjuncts is the possibility to specify cardinality constraints on role playing, determining for a natural type how many instances of a certain role type are related to it. Such constraints could not be specified graphically (and easily) in the roles as specializations approach, since a role type and a natural type are connected via generalization instead of association. Examples of "roles as adjunct instances" approaches that use this role cardinality constraint feature are (Wieringa et al., 1995) and (Cabot & Raventós, 2004). Further, the natural type association end should always be "read only" and its cardinality constraint should always be exactly 1, since a role object depends on exactly one and the same role player object (Cabot & Raventós, 2004).

According to (Steimann, 2000), "the modeling of roles as adjuncts remains practically appealing; in fact, it has been recognized as the only legitimate object-oriented implementation of roles". Nevertheless, the issue with the role as adjuncts approach is that authors do not usually consider the nature of roles as relationally dependent on entities external from their players. For example, Figure 4.6, taken from (Cabot & Raventós, 2004), represents an Employee role type, but does not represent an Employer role type; *mutatis mutandis*, Student and University, Project Manager and Project. Similar situations are found in (Wieringa et al., 1995) and in (Gottlob et al., 1996).

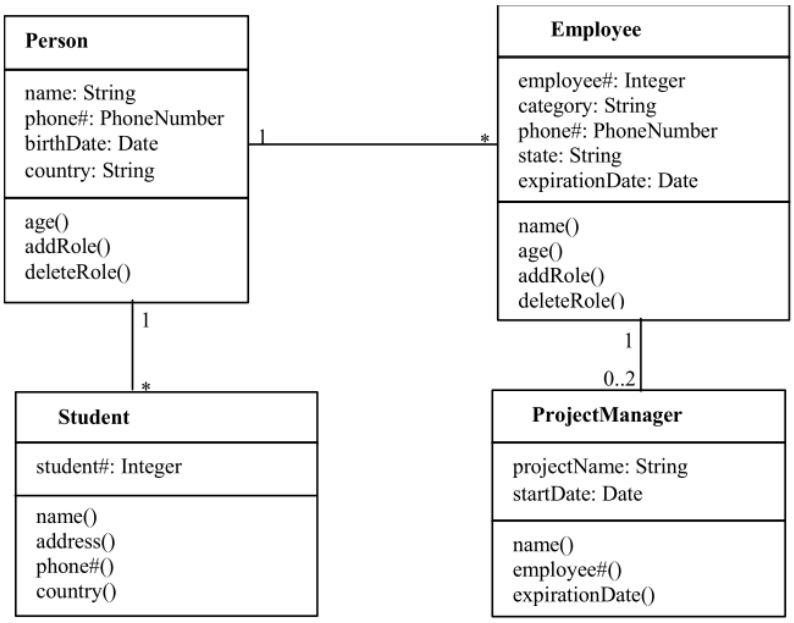

Figure 4.6 - Roles as adjuncts (without relational dependency) (Cabot & Raventós, 2004)

In our model-driven approach, roles at the ontological level are described as relationally dependent universals and, consequently, are represented in the context of material relations founded by relators. As a result, domain ontologies, which are used as our starting point, describe role playing in a more complete manner, which is not considered by the advocates of the "roles as adjunct instances" approach. Adopting the "roles as adjuncts" approach would only be suitable for addressing cases of partial information demand, when one is not interested in all aspects of role playing. Hence, we propose an approach that works more suitably in our model-driven context.

4.2.2 APPROACH

At the ontological level, particularly in OntoUML, Roles must be mediated by Relators, which are "aggregates of qua individuals" and must be explicitly represented in domain ontologies (Guizzardi, 2005). Furthermore, in OntoUML, "the externally dependent moments of a qua individual are represented as resultant moments of the relator" (Guizzardi, 2005). That is to say, properties of Roles are represented as belonging to Relators. At the information level, we must be able to address the case of full information demand on a role playing situation, one that involves interest in the Relator Universal and all the mediated Role Universals (as well as properties thereof).[11]

[11] Since in the adopted domain ontologies moments characterize relators (instead of roles), our information level approach will not be able to distinguish which *qua individual* actually bears a moment of a relator. Nonetheless, such distinction is possible, for example, when qua individuals are explicitly represented in domain ontologies.

As a result, we propose an approach that combines “roles as named places for relationships” with a variation of “roles as adjunct instances”. More specifically, we represent Relators as types and directly associate them with the natural types corresponding to the mediated roles. An example is depicted in Figure 4.7. In this approach, a relator type works as role types in the “roles as adjunct instances” approach (e.g., the Employment type acts as if it was both an Employee and an Employer type). To clarify that a natural type is playing a role in its association with a relator type, we include the role name in its association end, similar to the “roles as named places for relationships” approach.

Since a substantial may not necessarily play a role, the main object (i.e., the data fragment about static aspects of the substantial) may be unrelated to instances of the relator type. Consequently, this solution requires the use of optional cardinalities (lower bound 0) in the association ends attached to relator types. Moreover, the association ends attached to natural types should be “read only” (for the sake of simplicity, we omit the corresponding notation in the information models presented here).

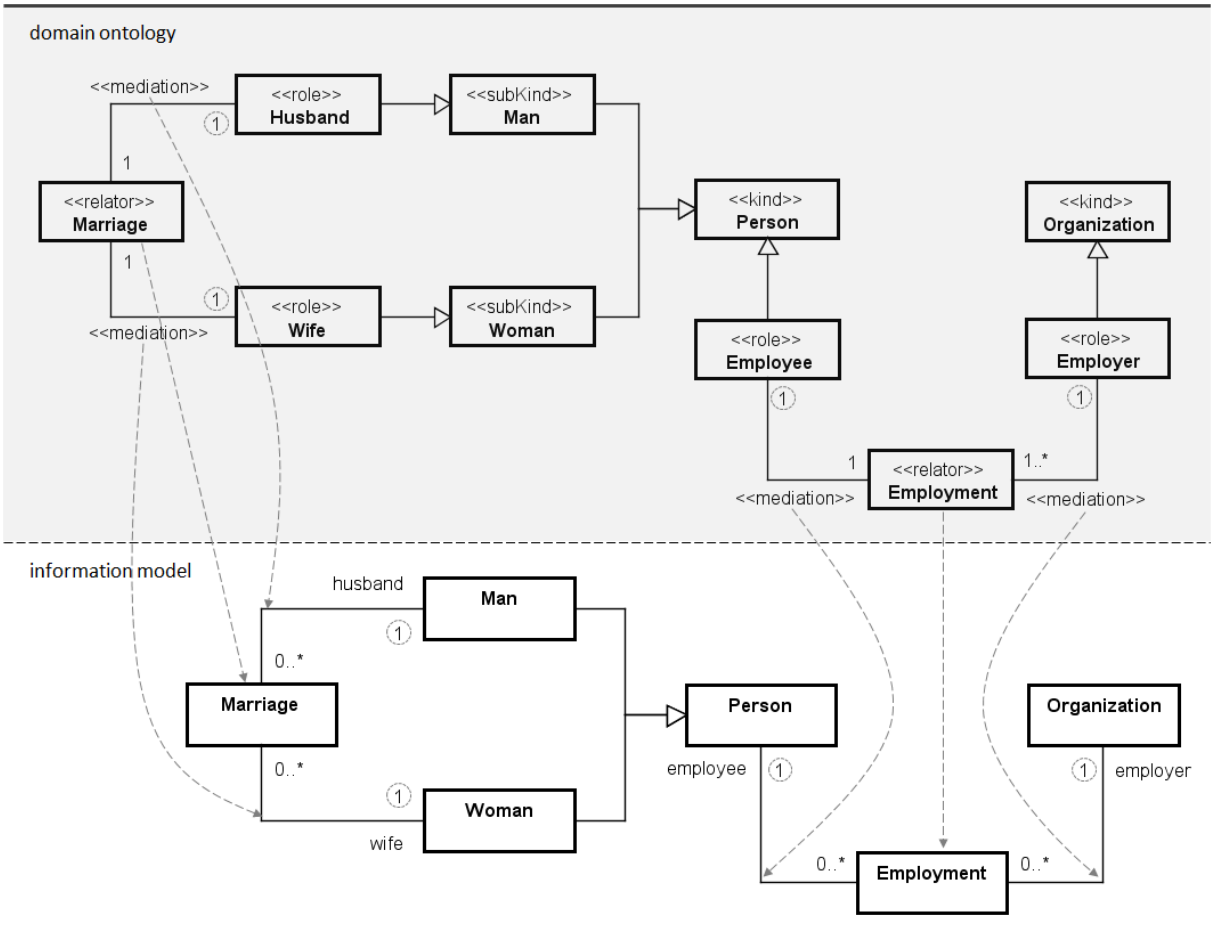

Figure 4.7 - Our information modeling pattern for Roles (via relator types and role associations)

Another point to be addressed is that substantials of different Kinds can play the same role. In the running example of Figure 2.8, this is the case for the Customer Role Mixin, which can be played both by people (private customers) and organizations (corporate customers). At the information level, using the "roles as adjunct instances" approach, this situation is usually solved by creating (or finding) a super type that is common to all natural types that play the role. For the customer example, this means that a super class of Person and Organization is used as a role player; this class is called Legal Entity in (Gottlob et al., 1996), Party in (Steimann, 2000) and Legal Person in (Cabot & Raventós, 2004). Then, the "roles as adjunct instances" pattern is simply applied to this super type.

Similarly, we do create a super type for the natural types that play the role. Henceforth, we call it the "mixin type". Then, we apply the same approach as we do on standard roles, i.e., we relate the mixin type with a relator type via association. An example of this pattern is depicted in Figure 4.8. In the information model, the mixin type is called "Potential Customer", although it would be better called "Legal Entity". We explain the issue in the following.

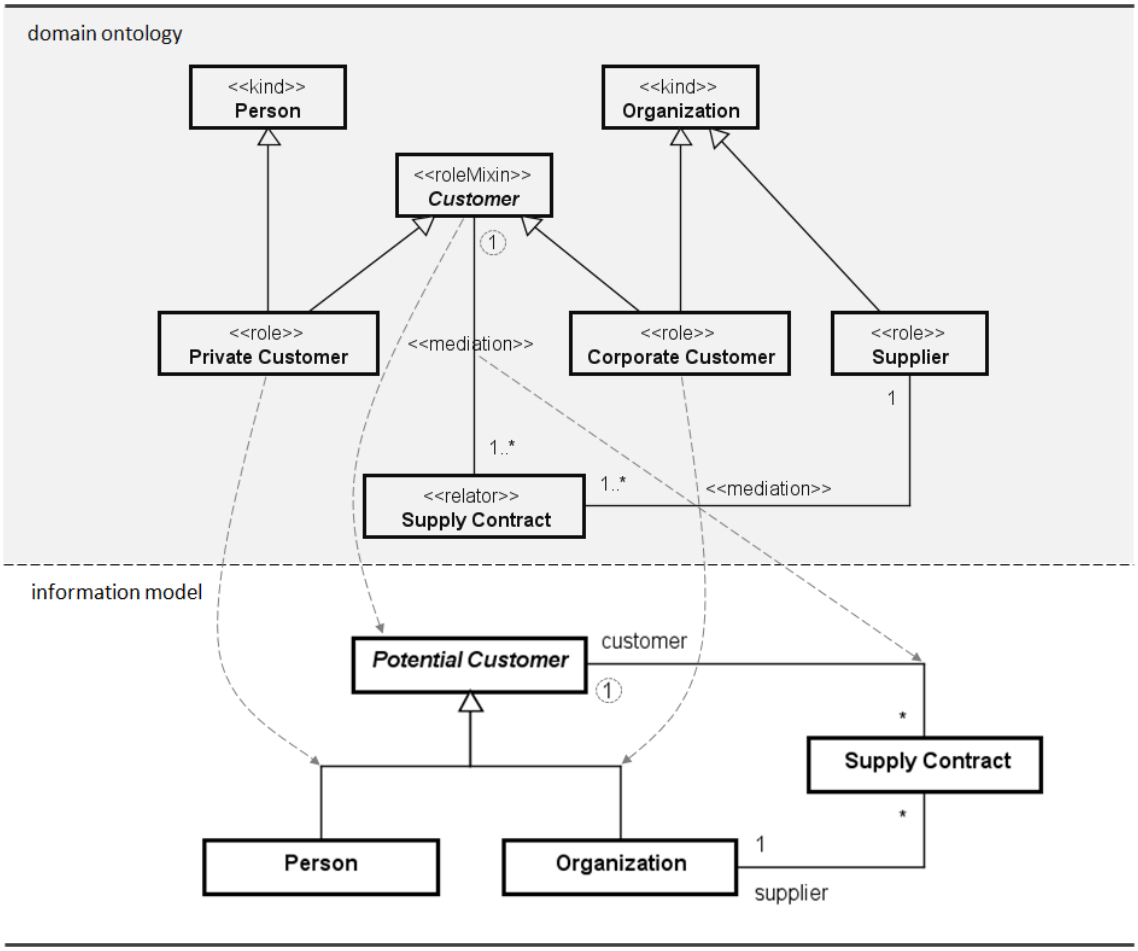

Figure 4.8 - Our information modeling pattern for Role Mixins (via mixin types)

In OntoUML, a Role Mixin (e.g., Customer) does not necessarily have to specialize a Category (e.g., Legal Entity). Neither the disjoint universals playing the Role Mixin (e.g., Person and Organization) are required to have a common Category ancestor (e.g., once more, Legal Entity). Hence, in our model-driven approach, we cannot programmatically create a suitable name for the mixin type (e.g., "Legal Entity").

We could name the mixin type after the Role Mixin's name (e.g., "Customer"). Nonetheless, such name is likely to have a role nature. This would wrongly suggest, for example, that every object that stores data about a person or an organization *always* stores data about the entity qua customer. Therefore, we name the mixin type after the Role Mixin's name plus the prefix "Potential" (e.g., "Potential Customer"). This would suggest, for instance, that every object that stores data about a person or an organization, *potentially* stores data about the entity qua customer. Notwithstanding, the name of the mixin type can be manually altered after the transformation.

4.3 CONCLUSIONS

Data about entities in reality is encapsulated in data fragments (objects) that instantiate types (classes). In our model-driven approach, the structure of types in an information model is built from the structure of universals in a domain ontology. Although all types perform the same function in an information model (viz. data syntax), we distinguish between three sorts of types, namely, natural types, mixin types and relator types. The distinction is merely made for explanation purposes, while illustrating the model-driven approach. Natural types are types corresponding to Kind and SubKind universals. Mixin types are types corresponding to Category (rigid Mixin) and Role Mixin (anti-rigid Mixin) universals. Relator types are types corresponding to Relator universals.

At the ontological level, particularly in OntoUML, the use of Role and Role Mixin universals (which are anti-rigid universals) relies on dynamic classification of entities in reality. At the information level, to avoid the use of dynamic classification in information models, we propose an approach that combines the approaches that Steimann refers to as "roles as named places for relationships" and "roles as adjunct instances". In our solution, we leverage an important characteristic of the representation of roles in OntoUML, namely, the emphasis of Roles as relationally dependent universals that occur in the context of Material Relations, which in turn are founded on Relators. That is to say, in OntoUML, a Role is represented along with a Relator and other Roles mediated by that same Relator (e.g., Employee, Employment and Employer).

At the information level, the relational dependency of roles is not usually considered by information and object-oriented modeling approaches, which usually represent roles outside of the context of material relations. Consequently, the previous approaches present roles as types. Because we are using OntoUML domain ontologies as a starting point, we instead represent relators as types.

Instances of a relator type carry data about entities in the context of a certain material relation (i.e., properties of entities in role playing and properties of relators).

Our representation of data on roles presupposes the usage of optional cardinalities in information models, which is admissible since we are representing *data* and not *reality*. At the ontological level, optional cardinalities are forbidden in OntoUML domain ontologies, since “from an ontological standpoint, there is no such a thing as an optional property” (Guizzardi, 2005).

In summary, for a certain entity in reality, data about static aspects is encapsulated in a single “main object” and data about dynamic aspects is separated in several “relator objects” that are related to the main object. The main object is an instance of a certain natural type (which may inherit from other natural types and mixin types), while each relator object is an instance of a relator type.

Hitherto, types (classes) bear no attribute, thus data fragments (objects) are acting as mere representatives for entities in reality. Nevertheless, on discussing further informational concerns (viz. history and time tracking, reference and measurement), types will possess attributes and will better provide the syntax of data about entities in reality. This will be addressed in chapters 6 and 7.

5 SCOPE

The informational concern of scope involves the selection of which categories of being (of a domain conceptualization) are relevant to an informational agent. In this chapter, we discuss how this concern impacts the syntax of data and, thus, the information model. Once more, we refer to fragments of the running example of Figure 2.8.

We assume that, for every universal in the domain ontology, there is an informational decision concerning whether it is included in the scope of the information demand. We identify decisions concerning dynamic aspects (which apply to Roles and Role Mixins) and decisions concerning static aspects (which apply to Kinds, SubKinds and Categories), which have different impacts in the structure of the information model. We discuss the dynamic aspects first (in section 5.1) since these have less impact on the structure of the overall model when compared to the static aspects (discussed in section 5.2).

In our explanations, we consider in each step a domain ontology fragment and initially present a corresponding information model capturing a full information demand (following the patterns of Chapter 4). Then, we present possible scope decisions along with the resulting information models. As a consequence, the structure of the resulting information models can be compared, taking the full information demand model as a point of reference.

5.1 DYNAMIC ASPECTS

On the one hand, at the ontological level, Roles are evidenced as relationally dependent universals that are instantiated in the context of material relations. As a result, role playing is specified in a complete manner, i.e., role playing requires the specification of the Relator universal, the involved Role universals and, optionally, the (derived) Material Relation universal. For instance, a marriage depends on a husband and a wife, an employment on an employer and an employee, a pregnancy on a pregnant woman and a fetus, a supply contract on a supplier and a customer.

On the other hand, at the information level, one can be nevertheless interested in role playing in a partial manner. For instance, one may be only interested if a woman is married (unconcerned about husband), if a woman is pregnant (unconcerned about the fetus) or if someone is currently employed (unconcerned about employer).

5.1.1 SCOPE OF ROLES AND RELATORS

In the following, we describe informational decisions concerning basic role playing, which are depicted in Figure 5.1. The universals considered outside the scope of the information demand are marked with an “X”.

The first pattern (Figure 5.1a) represents the situation of full information demand, where the Relator and all the Roles are in scope. As a result, the information model will have the corresponding relator type and role associations.

If a Role is not in scope, there will be no corresponding role association in the information model. This is illustrated in the second, third and forth patterns (Figure 5.1: b, c and d). On the second pattern, the agent is still interested in storing data about people, but is uninterested in data about people as employees. *Mutatis mutandis*, the third pattern considers organizations as employers as being outside the scope. On the forth pattern, data on people and organizations is no longer connected to data on employments, but one may still be interested in aspects of employments, such as counting and timing.

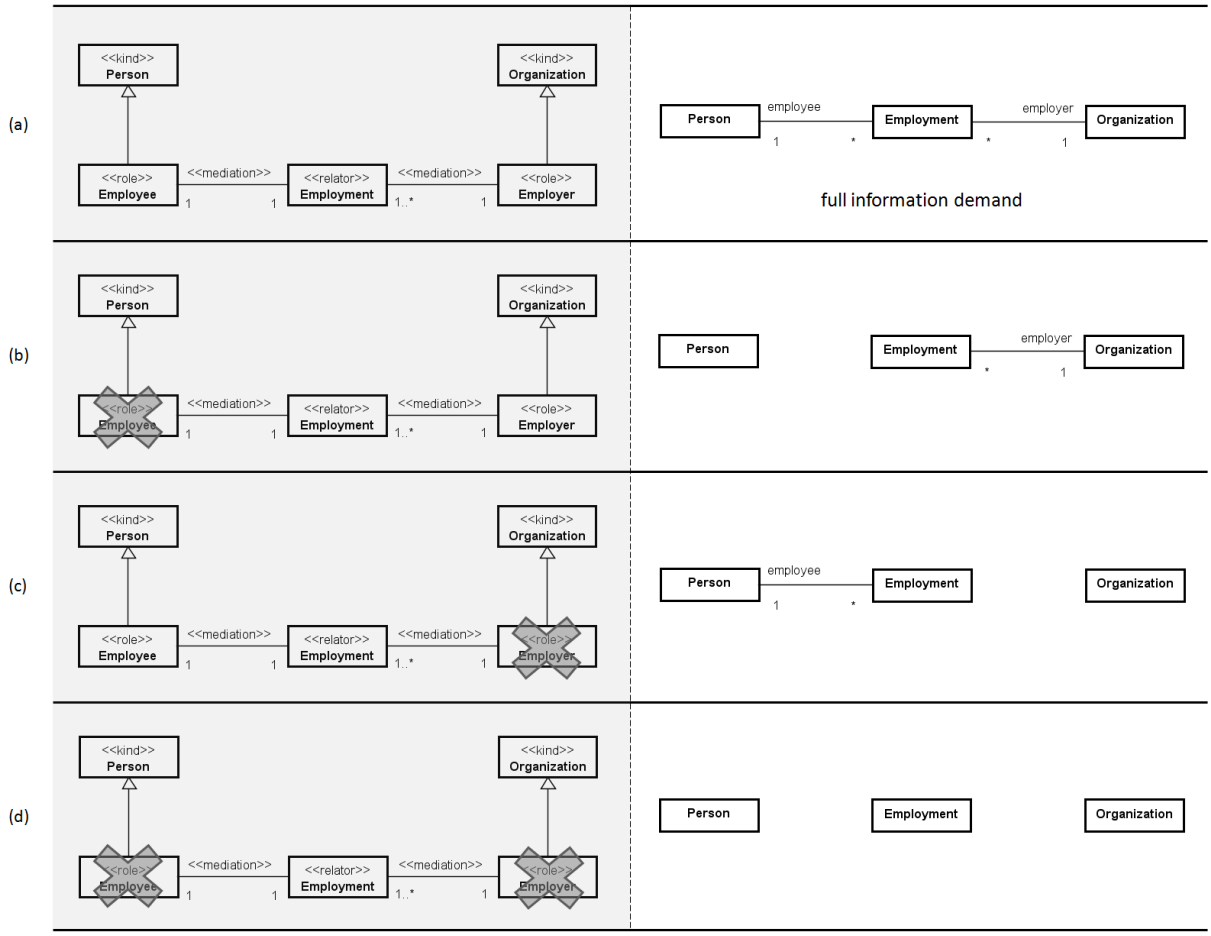

Figure 5.1 - Patterns for scope decisions on Roles

We assume that in order to be interested in a certain role playing aspect, the Relator that mediates the Roles must be in scope. Otherwise, the information model will not have the corresponding relator type and, consequently, role associations will be absent. This is illustrated in the pattern of Figure 5.2b, where one is completely uninterested in employments.

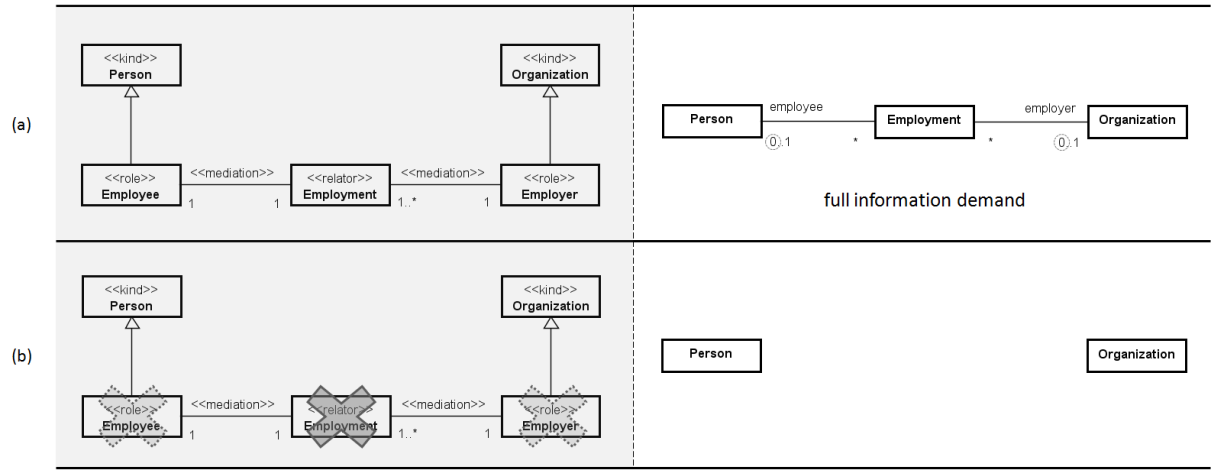

Figure 5.2 - Pattern for scope decisions on Relators

Finally, in role playing, another decision to be considered is that of relaxing cardinality constraints on role associations, on the natural type end. For instance, in the pattern of Figure 5.2a, cardinalities on the Person type end and on the Organization type end were relaxed. This means that, for data on a certain employment, data about the employer or the employee may be unknown. This may be applied to any role association, e.g., on the patterns of Figure 5.1b and c.

There are further scope reduction situations that we do not pursue here for the sake of simplicity. For example, an agent may be solely interested whether a substantial plays a certain Role universal or not (unconcerned about any properties and relationships). As an illustration, an agent may be interested whether a person is employed or not (unconcerned about employment(s), employer(s) and properties thereof). This would be typically addressed via a boolean attribute (e.g., “isEmployed”) owned by the corresponding natural type.

5.1.2 Scope of Role Mixins and Roles specializing Role Mixins

At the ontological level, in OntoUML, roles played by entities with different principles of identity are represented by a pattern composed of a Role Mixin, a Relator connected to it and the Roles that specialize the Role Mixin. As presented in chapter 4, the information level pattern for Role Mixins is slightly different from the one for Roles. Consequently, scope decisions on Role Mixins (as well as on Roles that specialize Role Mixins) cause a different impact on the structure of an information model. We present the patterns for Roles that specialize Role Mixins in Figure 5.3. The first pattern (Figure 5.3a) illustrates the full information demand case, which generates a mixin type corresponding to the Role Mixin and generalizations between the mixin type and the natural types representing the role players.

If a Role universal that specializes a Role Mixin is outside the scope, then the generalization between the corresponding natural type and the mixin type will be absent. This is illustrated in the second and third patterns (Figure 5.3: b and c). In the second pattern, the generalization between

the Person type and the Customer type is absent. As a consequence, objects (data fragments) that instantiate the Person type cannot be syntactically related to objects that instantiate the Supply Contract type. Analogously, the third pattern considers the Corporate Customer Role to be outside the scope and, as a result, the generalization between the Organization type and the Customer type is absent.

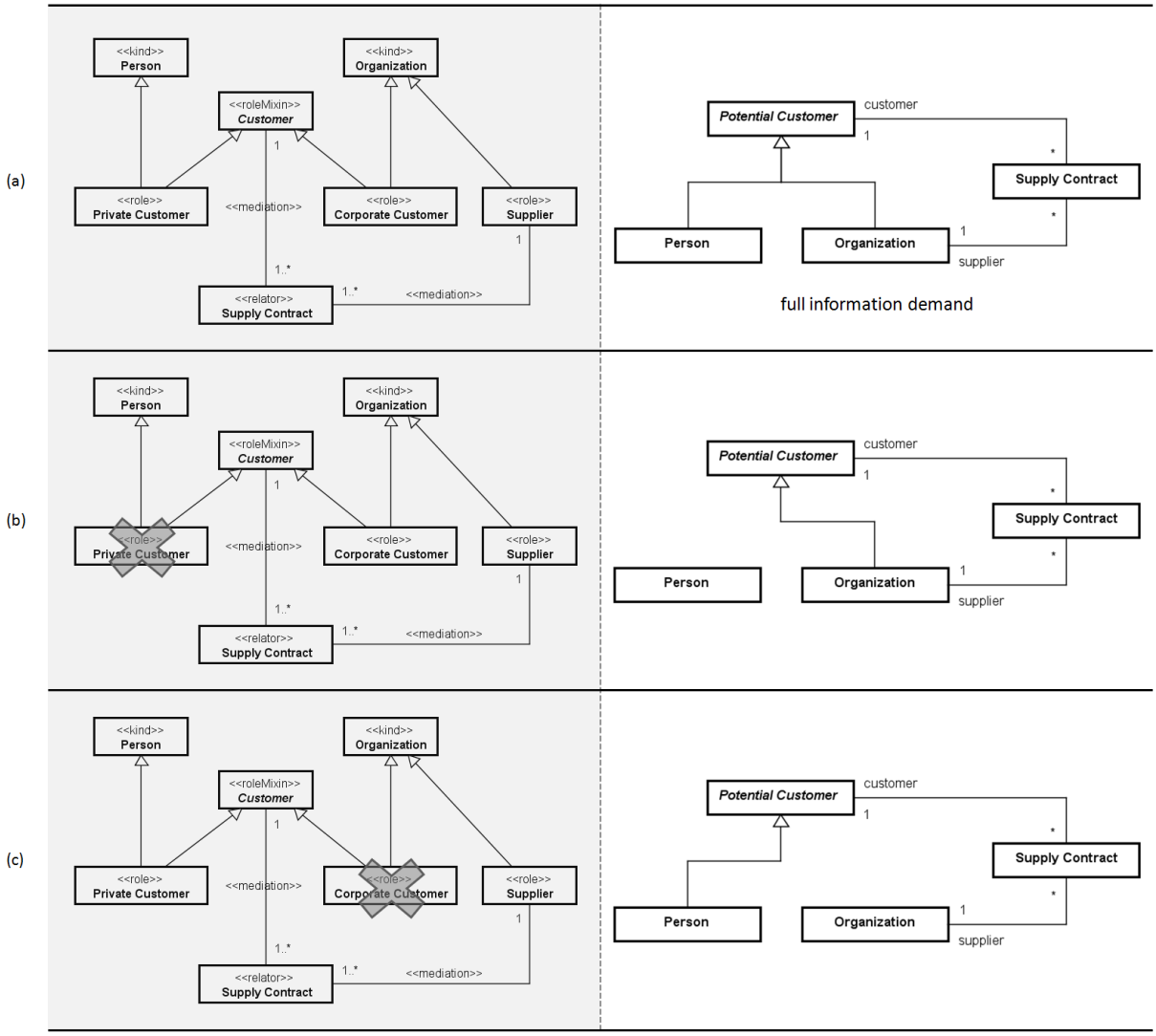

Figure 5.3 - Patterns for scope decisions on Roles specializing Role Mixins

We assume that in order for Roles that specialize Role Mixins to be in scope, the latter must be in scope as well. Otherwise, if the Role Mixin is outside the scope, the corresponding mixin type will be absent and so will be, as a matter of consequence, the generalizations between it and the natural types. This is illustrated in the pattern of Figure 5.4b. As with Roles, for a Role Mixin to be in scope, the corresponding Relator must be in scope. For instance, if the Supply Contract Relator is outside the scope, then the Customer Role Mixin will be outside the scope as well.

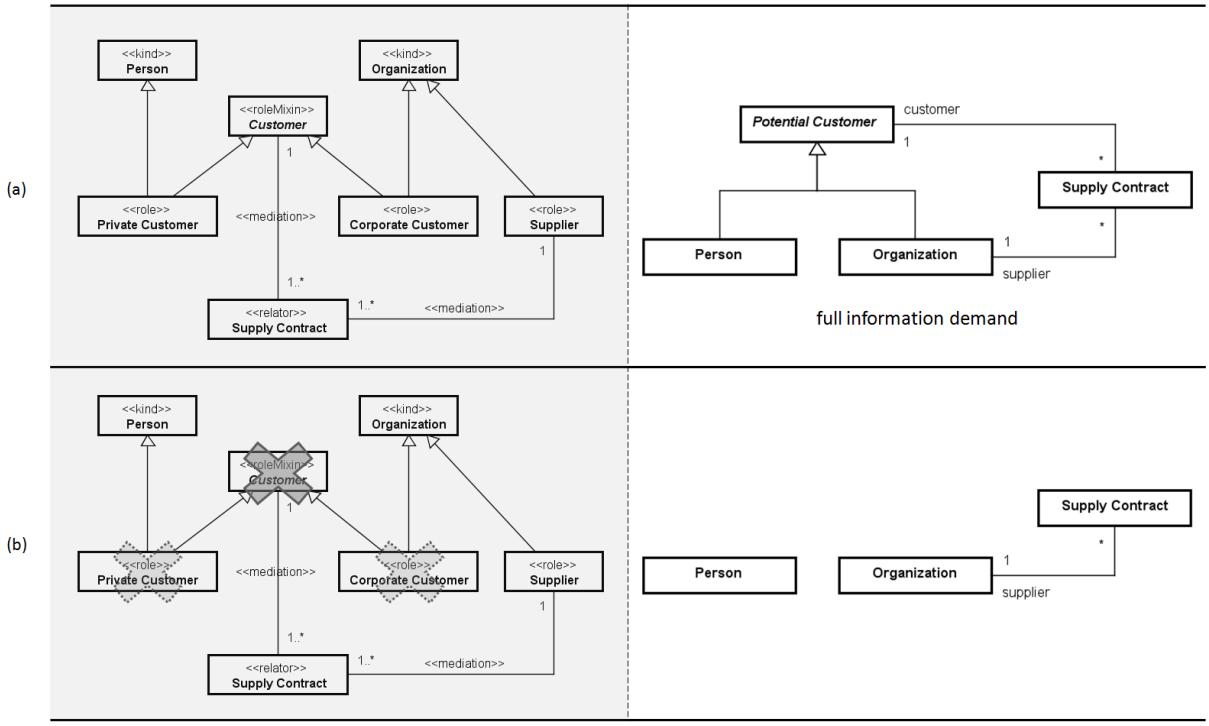

Figure 5.4 - Pattern for scope decisions on Role Mixins

5.2 Static Aspects

At the ontological level, Kinds (along with SubKinds) fundamentally determine what sorts of substantials are considered to exist according to a domain conceptualization. In addition, Categories represent abstractions of properties that statically apply to substantials of different Kinds. At the information level, an informational agent may be unconcerned about certain individuals that pertain to certain Kinds or SubKinds. Also, an agent may be unconcerned about classifying entities according to Categories. In this section, we initially discuss decisions on SubKinds, then Kinds and finally Categories.

5.2.1 Scope of SubKinds

At the information level, an informational agent may be exclusively interested in storing data about some specific SubKinds of a Kind. The patterns for scope decisions on SubKinds are depicted Figure 5.5. The first pattern (Figure 5.5a) illustrates the full information demand situation, in which natural types corresponding to the Man and the Woman SubKinds are created in the information model. Further, role associations attached to those natural types are created, according to the Role patterns, along with generalizations to the natural type corresponding to the Kind.

We consider that if a SubKind is outside of the scope, then the corresponding natural type will be absent in the information model. Due to the absence of the natural type, role associations and generalizations attached to it will also be absent. Consequently, if a SubKind is outside the scope,

then the specializing Roles will also be outside the scope. This is demonstrated in the second and the third patterns (Figure 5.5: b and c). In the second pattern, the Man SubKind is considered to be outside the scope and, consequently, the Husband Role. In the corresponding information model, the Man natural type is absent, as well as the role association to the Marriage type and the generalization to the Person type. Analogously, in the third pattern, the Woman SubKind is outside the scope, implying the same for the Wife and the Pregnant Roles.

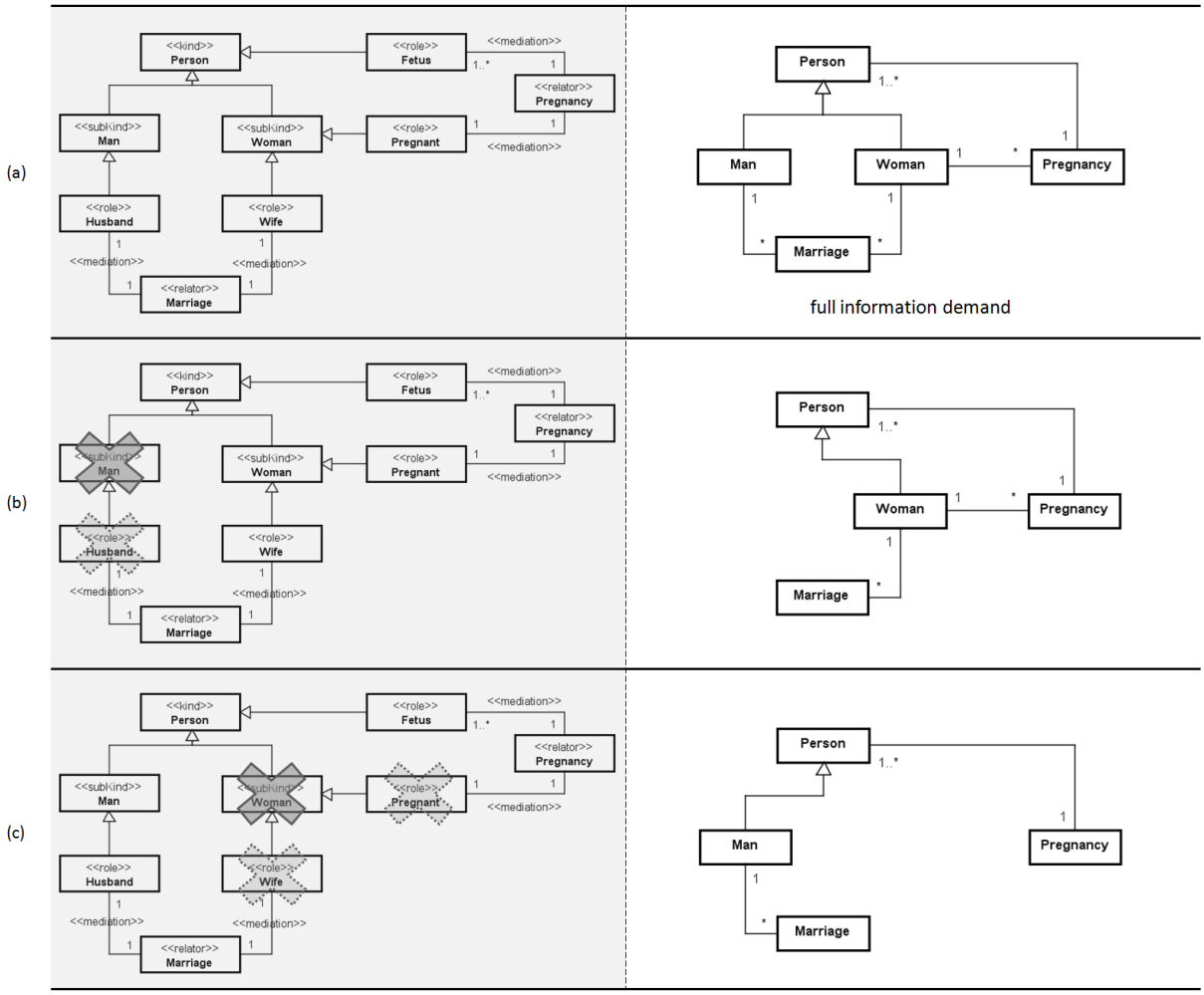

Figure 5.5 - Patterns for scope decisions on SubKinds

5.2.2 Scope of Kinds

At the information level, an agent may be completely uninterested in entities pertaining to a certain Kind. Figure 5.6 illustrates scope informational decisions on Kinds, which are analogous to those on SubKinds. The first pattern (Figure 5.6a) presents the case for full information demand, where the natural types corresponding to the Person and Organization Kinds are created. Furthermore, each natural type is related to other types (via association or generalization) due to the patterns for Roles, Role Mixins and Categories.

We consider that when a Kind is outside the scope, the corresponding natural type will be absent in the information model, as well as all the associations related to it. As a result, a Kind outside the scope implies that all its specializing Roles will likewise be outside the scope. We illustrate this case in the second and third patterns (Figure 5.6: b and c). In the second pattern, the Person Kind is considered outside the scope and, consequently, the Employee and the Private Customer Roles are also considered outside the scope. Hence, the generated information model does not provide syntactical constructs to record data about people. Analogously, in the third pattern, the Organization Kind is considered outside the scope, implying the same for the Employer, the Corporate Customer and the Supplier Roles.

Besides that, we also assume that if a Kind/SubKind is outside the scope, then all SubKinds specializing it are likewise outside the scope, causing a recursive pattern of scope reduction. For instance, in Figure 5.5, if the Person Kind is outside the scope, then the Man and the Woman SubKinds are also outside the scope. As a result, the scope of a Kind/SubKind implies the scope of all the further specializing universals, i.e., SubKinds and Roles.

It is worth to notice that this is not the only possible approach, i.e., we committed to specific design decision. Take for example, the information models of Figure 5.5. If the Person Kind is outside the scope, it is still possible to maintain the natural types for Man and Woman in the corresponding information model. Nonetheless, by doing so, the common link between objects that instantiate the Man type and objects that instantiate the Woman type is lost. Additionally, the Person type could originally possess a number of attributes (e.g., reference attribute, measurement attributes, history and time tracking attributes) and relations (e.g., role associations, generalizations). Consequently, if the Person type is absent in the information model, all those attributes and relations, common to the Man type and the Woman type, would be lost. To avoid this, attributes and relations of the Person type could be duplicated on the Man and Woman types; nevertheless, this would cause a considerable impact in the information model. Ergo, in this thesis, we have chosen a simpler approach for the scope of Kinds and SubKinds.

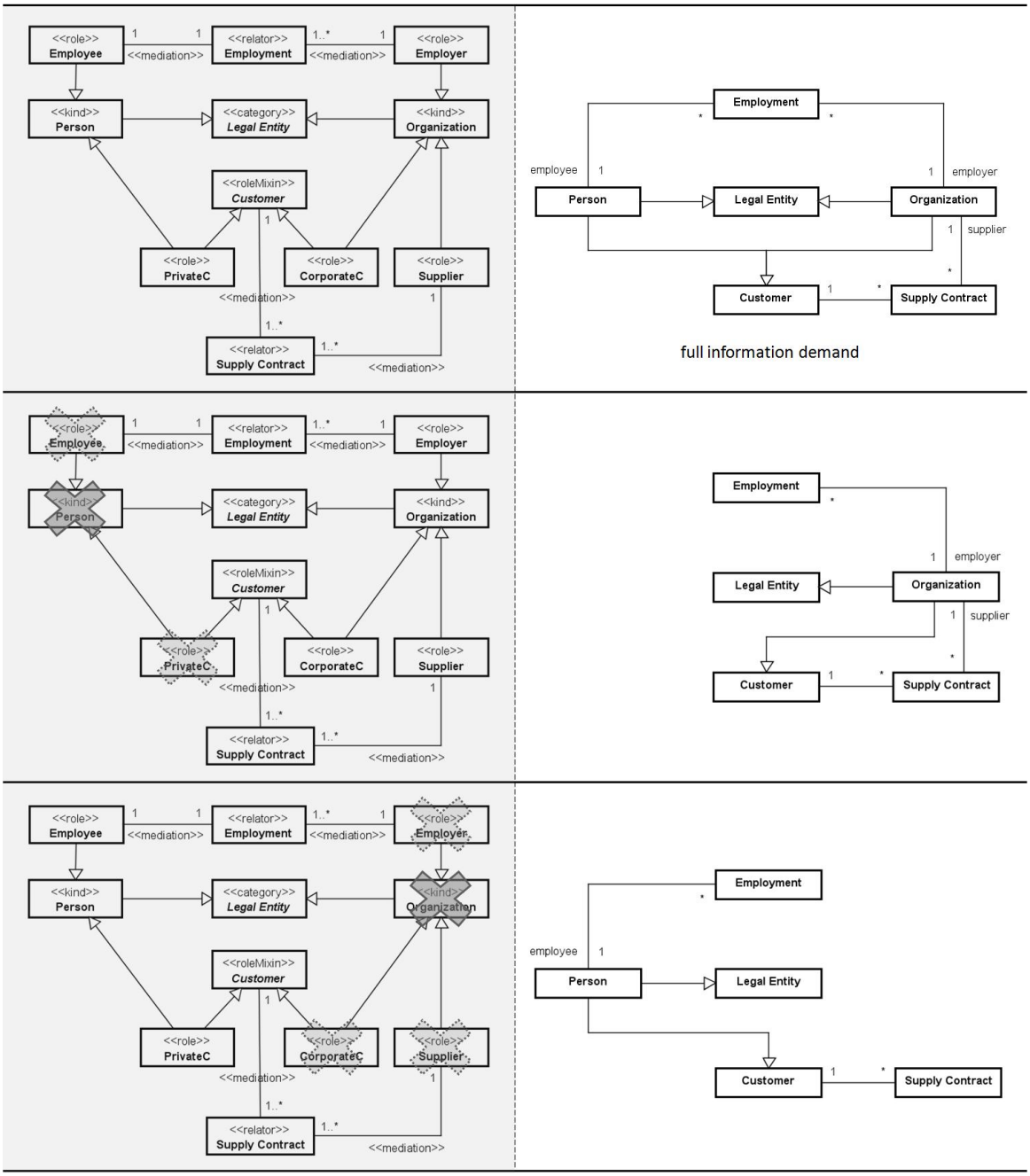

Figure 5.6 - Patterns for scope decisions related on Kinds

5.2.3 Scope of Categories

At the information level, an agent may be interested in entities that pertain to a certain Kind/SubKind, but unconcerned about their classification in terms of Categories. The pattern for scope decisions on Categories is illustrated in Figure 5.7. The first pattern (Figure 5.7a), describing a full information demand, presents the mixin type corresponding to the Legal Entity Category. We assume that a mixin type corresponding to a Category may possess attributes related to the Category and may also serve to aggregate data fragments (objects) of different types.

When a Category is outside the scope, the only impact produced in the information model is that the corresponding mixin type will be absent. This is depicted in the second pattern (Figure 5.7b). We consider that a Category outside the scope does not imply that the specializing Kinds, SubKinds and Roles will be outside the scope. We do such on the premise that the fundamental decision to track entities depends on the scope of Kinds and SubKinds, as previously seen.

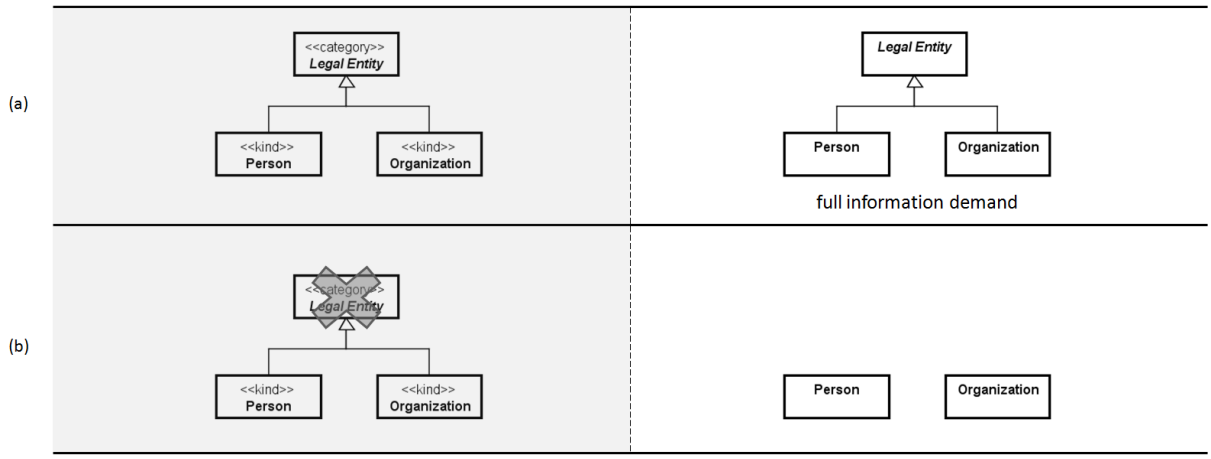

Figure 5.7 - Pattern for scope decisions on Categories

In essence, our approach concerning the scope of Kinds/SubKinds and Categories is founded on the theories of the ontological level. At the ontological level, a substantial must obey a *unique* principle of identity supplied by a *unique* Kind. Then, such principle of identity is carried by specializing universals (viz. SubKinds and Roles). Moreover, a substantial may be classified according to *several* Categories. Each Category provides a principle of application, but lacks a principle of identity (Guizzardi, 2005). For those reasons, we consider that the scope of a Kind/SubKind determines whether a substantial is inside or outside the scope of the information demand, while the scope of a Category is limited to keeping track of certain properties of the substantial (namely, those that are associated with the principle of application provided by the Category).

5.3 CONCLUSIONS

There are several patterns for scope decisions, which concern different meta-categories (such as Kinds and Roles) and cause different impacts in the resulting information model. Furthermore, scope decisions on certain universals may affect the scope of other universals.

We present a summary of this chapter in Table 5.1. Each line describes the impact of considering a universal, of a certain meta-category, outside the scope. The first column presents the meta-category of the universal that is the target of the scope decision. The second column shows possible "side effects" produced by considering the universal outside the scope. Finally, the third column lists the constructs in the information model that will be absent, as a direct consequence of the universal being considered outside the scope. In this column, we highlight the main construct to be affected; other constructs are absent as a result of the absence of the highlighted one.

Meta-category of the universal outside the scope	**Meta-category of universals outside the scope ("side effect")**	**Absent constructs in the information model**
Kind	**SubKinds and Roles** specializing the Kind	- The corresponding **natural type** (consequently, generalizations to mixin types)
SubKind	**SubKinds and Roles** specializing the SubKind	- The corresponding **natural type** (consequently, the generalization to natural type (Kind) and generalizations to mixin types)
Role	–	- The corresponding **role association**
Relator	**Roles and Role Mixins** mediated by the Relator	- The corresponding **relator type**
Role specializing Role Mixin	–	- The **generalization** from the corresponding natural type to the corresponding mixin type
Role Mixin	**Roles** specializing the Role Mixin	- The corresponding **mixin type** (consequently, the role association and generalizations from natural types)
Category	–	- The corresponding **mixin type** (consequently, generalizations from natural types and generalizations from mixin types (Category))

Table 5.1 - Summary of scope decisions

6 HISTORY AND TIME TRACKING

History tracking concerns whether an informational agent is required to know about past and/or present things. Time tracking concerns whether one is required to know about the temporal extension of events concerning the existence of things.

In this chapter, we discuss history tracking followed by time tracking. For each of these informational concerns, we use the meta-categories of the ontological level to discuss what sorts of things can be considered as targets of history and time tracking. That is to say, we begin by identifying informational decisions. Afterwards, we describe our model-driven approach to address the informational concern, based on the identified informational decisions. In the end, we present concluding remarks.

6.1 HISTORY TRACKING

The world constantly changes: things begin and cease to exist. Things that currently exist may not be the only targets of information demand. In many circumstances, one is interested in things from the past, i.e., substantials and relators that ceased to exist.

6.1.1 INFORMATIONAL DECISIONS

We consider that, for every Relator universal in the domain ontology, one has a decision about which relator individuals are relevant to keep track of. This decision is two-fold: one has to decide if the past relators are relevant and also decide if the current relators are relevant. As a consequence, one may be interested in: (i) exclusively past relators (or a selection thereof), (ii) exclusively current relators, or (iii) both sorts of relators. For instance, an air traffic controller may only be interested in current flights, while an airline administrator may be interested in the history of flights.

We also consider that history tracking decisions target every Kind universal in a domain ontology. Similarly to the case of Relator universals, a history tracking decision for a Kind universal is two-fold and may result in tracking: (i) exclusively the past, i.e., the substantials that have ceased to exist, (ii) exclusively the present, i.e., the currently existing substantials, or (iii) both the present and the past, i.e., substantials that have ceased to exist and those that exist in the present.

Determining the circumstances in which a substantial begins and ceases to exist is a competence of the ontological level and is captured in the principle of identity of a Kind universal. Therefore, one should be aware of the definition of existence adopted for a certain Kind universal before committing to a history tracking decision. Consider for example the information demand of a cemetery which is interested exclusively on maintaining records of the deceased. Varying the principle of identity associated with Person in the domain ontology directly affects the history

tracking decision concerning people. If the domain ontology associates the concept of Person to the notion of living organism, people cease to exist when they die (viz. they stop being in space-time and only their body, which is a different concept, remains). In this case, one is interested solely on the past (people that no longer exist according to the ontology). However, if the domain ontology adopts a principle of identity for the concept of Person that is based on a theological or spiritualistic perspective, then people never cease to exist (viz. when people die, they become some sort of everlasting disembodied souls). In this case, one is interested in a portion of the existing people (namely, those that have passed away).

For Quality universals, history tracking can be done in terms of their assumed qualia (values). Alternatively, this could be seen as the history tracking of measurement events. For each Quality universal, it is a decision whether to keep only its current quale value (the last measurement event) or a history of values. For example, consider a car dashboard where one can see the various values assumed by qualities of a car, e.g., fuel level, speed, coolant temperature and odometer distance. An on-board diagnostic system may be interested in the current values of all those qualities but only some of them may actually require history tracking. We elaborate on history tracking of Qualities in chapter 7.

6.1.2 Model-driven Approach

Foremost, we discuss the possible solutions for full information demand, i.e., interest on both present and past things.

One possible solution is to divide objects (data fragments) into past and present via subtypes. That is to say, for each natural or relator type in the information model, there would be two subtypes, one representing data about past things and the other data on present things. For example, for the Person type, there would be a Past Person subtype and a Present Person subtype. The problem with this approach is that it relies on dynamic classification, since an object instantiating a present type may have to be reclassified as an instance of a past type. For example, an instance of the Present Person type may, later on, instantiate the Past Person type. As previously advocated, we avoid dynamic classification and, thus, we refrain from addressing history tracking via subtypes.

Hence, we adopt an approach that addresses history tracking via attributes. In full information demand, a natural or a relator type will own a boolean attribute that indicates whether an instance of the type conveys present or past information. We establish a convention in which we name such attribute “current” and a true value represents present information, while a false value represents past information. In the case of partial information demand (i.e., interest on either present or past), a natural or a relator type will not own a “current” attribute. In this case, instances of the type either convey past or present information, so the distinction between present and past instances, provided

by the “current” attribute, is unnecessary. The patterns for history tracking decisions on Kinds are illustrated in Figure 6.1. For a given Kind in a domain ontology, one may be solely interested in present instances (Figure 6.1a), past instances (Figure 6.1b), or both present and past instances (Figure 6.1c). When interest is exclusively on past, we add the prefix “Past” in the type’s name (e.g., “Past Person”) to emphasize that instances of the type are data on *past* substantials.

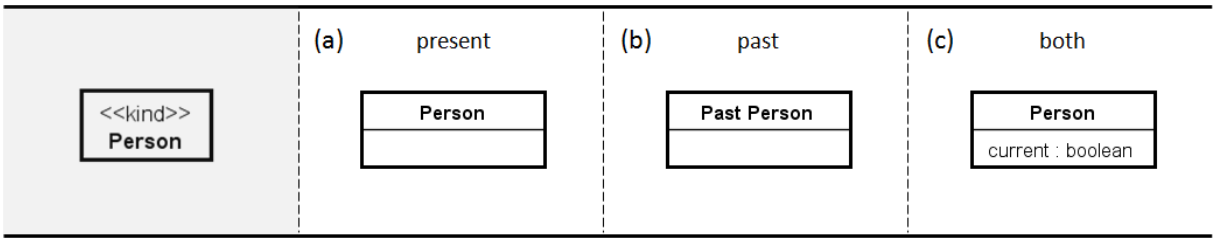

Figure 6.1 - Patterns for history decisions on Kinds (via “current” attribute)

For history tracking decisions on Relators, not only relator types are affected but also role associations, more specifically, the upper bound on the relator type end (henceforth, “relator type cardinality”). A relator type cardinality specifies, for an instance of a natural type (the main object), how many instances of a relator type (relator objects) may be related to it, at maximum. The minimum cardinality is unaffected, since the role playing pattern requires that it must be zero (i.e., roles are always optional for natural types).

We illustrate the patterns for history tracking decisions on Relators on Figure 6.2 (relator type cardinalities have been highlighted in the information models). For the sake of illustration, we assume that all three patterns (Figure 6.2: a, b and c) commit to a history tracking decision on present people and organizations, while varying history decisions on employments only.

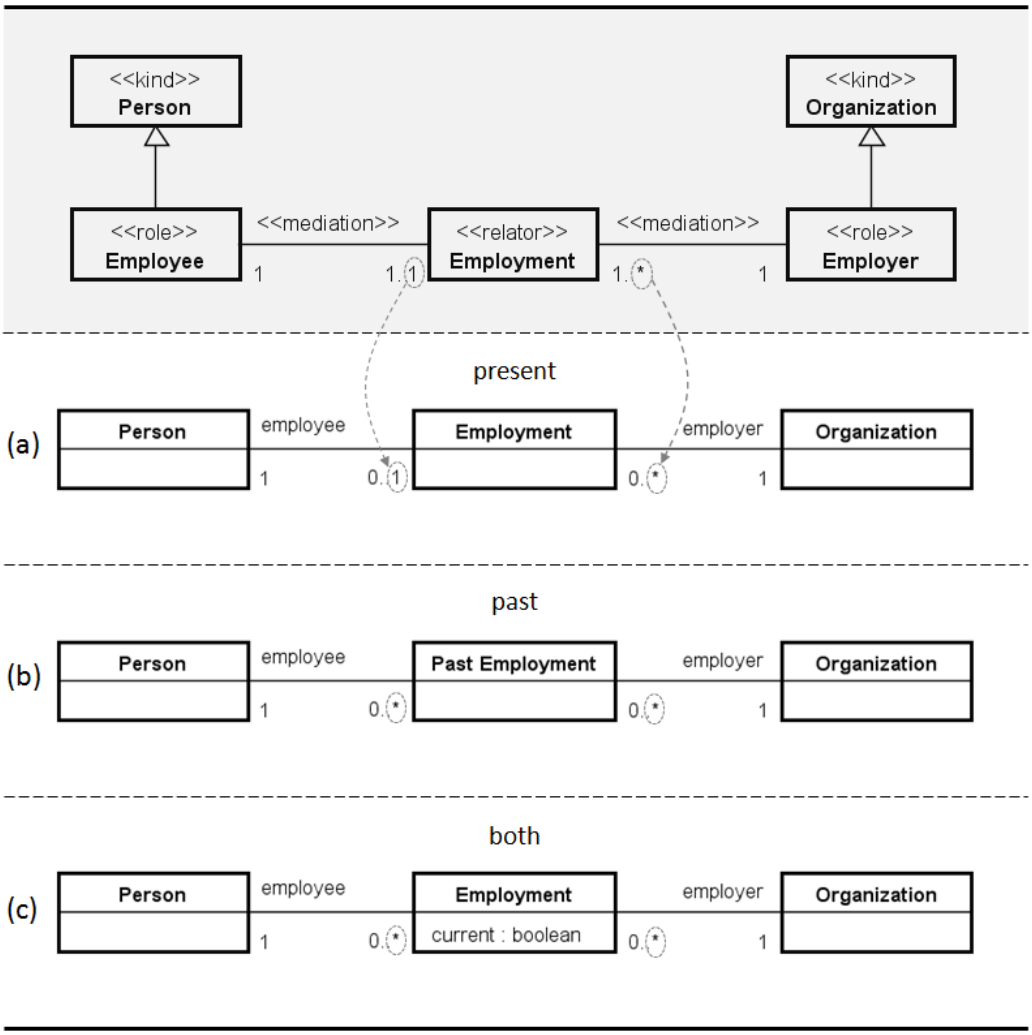

Figure 6.2 - Patterns for history decisions on Relators (via "current" attribute and relator type cardinalities)

The patterns were designed based on the following premise. We assume that the only thing specified in the domain ontology is how many times a certain substantial may play the same Role universal *simultaneously*, not throughout lifetime. This is specified in a mediation relation between a Role and a Relator universal, more specifically, in the upper bound of the Relator universal end (henceforth, "relator universal cardinality"). Cardinalities of this sort have been highlighted in the domain ontology of Figure 6.2. According to our assumption, a domain ontology does not systematically specify how many times a substantial may play a Role universal throughout lifetime. As an illustration, in the running example (Figure 2.8), it is not explicitly specified that a person may play the Fetus role only *once in a lifetime* and may play the Employee role unrestrictedly.

As a result, when interest is exclusively on present relators (Figure 6.2a), we take the relator type cardinality to be the same as the relator *universal* cardinality in the domain ontology. When interest is exclusively on past relators (Figure 6.2b) or on both present and past relators (Figure 6.2c), we take the relator type cardinality to be unrestricted ("*"). Thus, in the information models of Figure 6.2b and Figure 6.2c, instances of the Person type, as well as instances of the Organization

type, may be related to possibly many instances of the Employment type. Particularly, in the information model of Figure 6.2c, some of the instances of the Employment type may be data on past employments (“current” attribute equals to false) and some may be data on present employments (“current” attribute equals to true). Ergo, we take a permissive approach when dealing with information models that capture an information demand on past. By taking the relator type cardinality to be unrestricted, valid information models are created both in situations where the role may be played unrestrictedly throughout lifetime (e.g., Employee) and restrictively (e.g., Fetus). For roles that possess lifetime restrictions (e.g., Fetus), one could manually change the relator type cardinality (e.g., from “*” to “1”) after the model transformation.

Besides that, information models produced by our approach could be manually constrained to forbid invalid combination of data fragments. For example, when interested on past and present relators (Figure 6.2c), one could restrict the number of (data on) *current* employments that are related to (data on) a person.

When considering two history tracking decisions, one for a given Kind and other for a given Relator, there are several possible combinations. For example, consider the combinations of decisions concerning Person and Employment (for the sake of simplicity, consider Organization outside the scope). In Figure 6.2, by fixing a “present” decision on Person while varying decisions on Employment, we already illustrated three possible combinations.

Some of the remaining combinations are illustrated in the information models of Figure 6.3. In Figure 6.3a, the information demand is exclusively on past. In Figure 6.3b, the information demand is on present and past people and solely on past employments. Both of the aforementioned information models only specify *valid* combinations of (data on) Kinds and Relators, since (data on) past employments may either be combined with (data on) present or past people. However, when combining history decisions, some information models may have to be further constrained, if one wishes to avoid the *invalid* combination of (data on) *present* employments with (data on) *past* people. This is the case for the models of Figure 6.3c and Figure 6.3d. Those models address an information demand on past and present people. The former model is solely interested on present employments and the latter model is interested on both present and past employments.

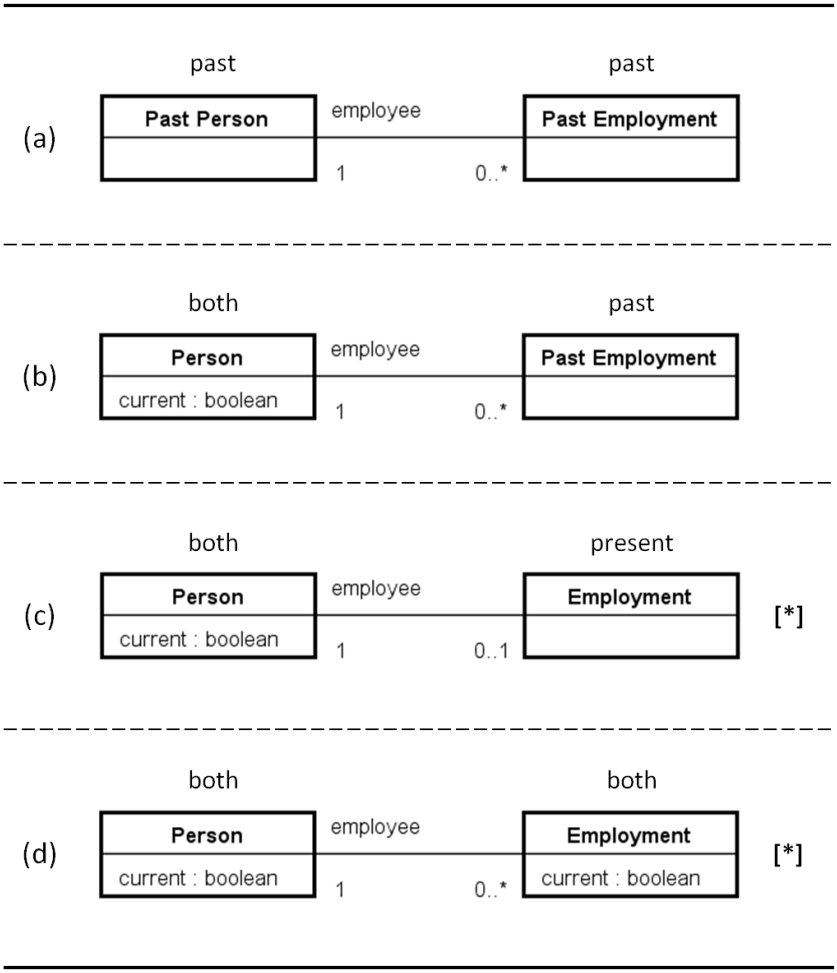

Figure 6.3 - Some combinations of history decisions on Kinds and Relators

6.2 TIME TRACKING

Although domain ontologies considered here are exclusively focused on structural aspects of reality, *perdurants* (events) play a central role in discussions on time tracking. We consider that an important built-in feature of the ontological level is that the lifecycle of each endurant is encompassed by an implicit event of existence. Such feature is acknowledged in the Basic Formal Ontology (BFO) foundational ontology: “One type of relation is of special importance (…) this is the relation between each [endurant] and that unique [event] which is its life” (Grenon, Smith, & Goldberg, 2004) (terminology has been adapted). The event of existence, as any event, has timing aspects such as start and end time, as well as duration. We consider those basic timing aspects to be built-in features of the ontological level, since by definition all (non-atomic) events possess them. Albeit those aforementioned features are implicit at the ontological level, one has to inevitably address issues on history and time at the information level, as those are likely to affect the structure of data.

6.2.1 Informational Decisions

In general, time tracking concerns the interest on timing aspects of things, i.e., properties corresponding to the time dimension. Here, we focus on events underlying the existence of Kind and Relator universals, particularly, on three basic timing aspects: (i) start time of events (e.g., date of admission in a job), (ii) end time of events (e.g., graduation year), and (iii) duration of events (e.g., duration of a phone call).

As a result, for every Kind and Relator universal in the domain ontology, one has the informational decision of tracking the relevant basic timing aspects. For instance, consider a domain ontology specifying enrollments as relators. One informational agent may be interested exclusively on the start time of enrollments (e.g., to check for current students that are about to exceed the maximum duration of a course). Another agent may be interested exclusively on the duration of past enrollments (e.g., to know how long past students took to graduate). Besides that, a system could be interested in none of the basic timing aspects (e.g., a librarian could be interested if someone is enrolled or not, but unconcerned about any timing aspect of enrollments).

Although we only deal here with basic timing aspects, there are other timing aspects that may also be relevant to information modeling. For example, one may be interested in the frequency of events (e.g., how many accidents have occurred in the last year), prospective information (e.g., when certain events are expected to occur) and derived information (e.g., age). These are considered outside the scope of our approach.

6.2.2 Model-driven Approach

For time tracking, the transformation pattern is straightforwardly done via attributes. Since time tracking decisions are related to Kinds and Relators, those decisions will affect their corresponding types, namely, natural and relator types. A decision to time track a certain universal results in a time attribute in the corresponding type.

The attributes for start time, end time and duration, by convention, are named “start”, “end” and “duration”, respectively. We assume that the attributes for start and end time are of a DataType that we call, by convention, “TimeInstant”. For the duration attribute, we assume it is of the DataType that we call “TimeInterval”. Here, we do not commit to any syntactical specifications for those DataTypes, but rather leave them to be suitably defined and agreed by the users of the information model (a measurement concern). As illustration, time instants may be encoded in the “yyyy/mm/dd hh:mm” format (e.g., “2012/04/08 3:47”) and time intervals may be in minutes (e.g., “7” minutes). We include both DataTypes in the information model.

When registering timing information on substantials and relators, one will not initially have information on the end time of things that currently exist. Thus, when there is interest on present, we specify the end attribute as optional ("[0..1]"). An example of full information demand on time tracking is depicted in Figure 6.4.

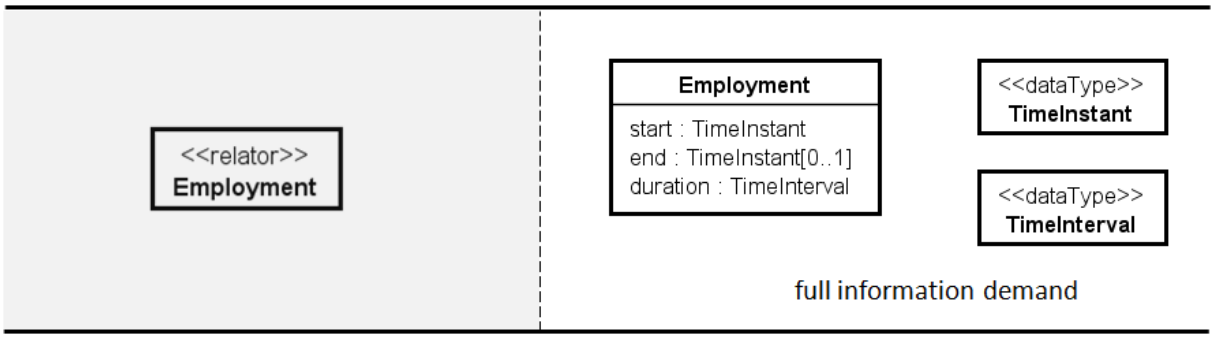

Figure 6.4 - Pattern for time tracking decisions (via attributes)

For a given natural or relator type, it is possible to compute values for one time attribute, given values on the other two. For instance, given values on start and end times, it is possible to compute values on duration. Although we have not taken derivation of time attributes into account in our model-driven approach, the user may still customize the resulting information model after the transformation. As a result, the user may indicate which attributes are derived and write derivation rules. For instance, consider the information model of Figure 6.5, committed to history tracking decisions on past. In this model, the "end" attribute of Past Person and the "duration" attribute of Past Employment were set as derived (derivation rules have been omitted). If derivation is not used in the group of three time attributes, integrity rules may be written to enforce that the three stored values are in conformance. This could be the case for the information model of Figure 6.4.

Additional constraints may be considered when combining time tracking decisions for a Relator and the involved Kinds. As accounted by the ontological level theory, role playing situations occur within the lifecycle of substantials playing the roles. Consequently, the lifecycle of a relator is within the boundaries of the lifecycle of the involved substantials. Accordingly, in information models, time attributes of a relator type should conform to time attributes of the related natural types. As an illustration, reconsider the information model of Figure 6.5. For a given (data on) past employment, time values have to be within the boundaries of time values of (data on) the related person and organization.

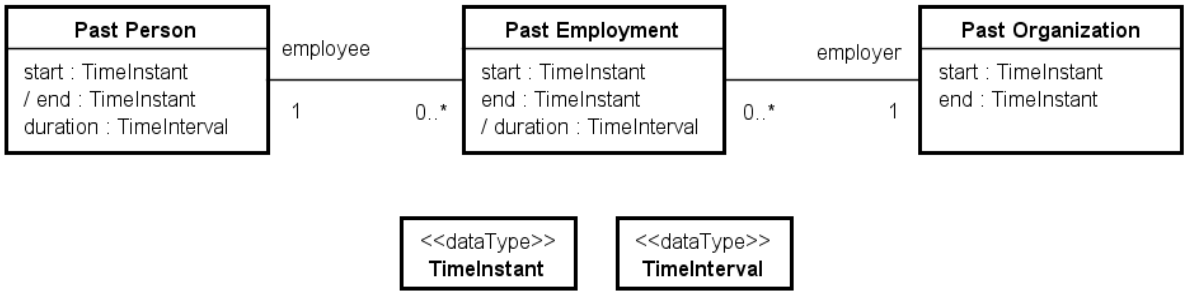

Figure 6.5 - Derivation of time attributes and combination of time tracking decisions

6.3 CONCLUSIONS

In this chapter, we have seen that the proper addressing of both history and time tracking, at the information level, require a deep understanding of some ontological aspects, namely, the circumstances in which things begin and cease to exist. We assume here that there are several aspects *underlying* the specification of a domain ontology. In particular, we consider that the theories at the ontological level grant that things begin and cease to exist and their lifecycle is encompassed by implicit events of existence. At the information level, nevertheless, one has to inevitably decide which things are of interest (in terms of past and present) and which attributes of the implicit events of existence are relevant. After all, as we depicted, decisions on history and time tracking affect the structure of information models. More specifically, in our approach, history tracking may require the use of a “current” attribute and also affects what we called “relator type cardinalities”. In addition, time tracking decisions involve the use of time attributes (“start”, “end” and “duration”) and data types.

7 Reference and Measurement

This chapter is devoted to two distinct informational concerns, namely, reference and measurement. Although both concerns are usually addressed via attributes in information models, they have different nature. The distinction between reference and measurement should be evidenced as we elaborate on informational decisions and present our model-driven approach for each of those concerns.

This chapter is structured as follows. In section 7.1, we investigate how we refer to individuals in reality. Our investigation is supported by a topic in philosophy called *reference*, concerned with the relation between "expressions and what speakers use expressions to talk about"[12]. In section 7.2, we elaborate on the measurement of properties pertaining to individuals in reality. In that section, we take specific care to contrast measurement with reference. In section 7.3, at last, we present concluding remarks.

7.1 Reference

7.1.1 Introduction

In this thesis, we are specifically interested in reference to *individuals*; not universals or other sorts of things. According to (Strawson, 1950), we refer to individuals using certain "classes of expressions". When the speaker has direct access to the thing being referred, he can use "ostension" – i.e., the act of pointing to something to refer – along with demonstrative pronouns ("this" and "that"). In the context of a particular conversation, personal and impersonal pronouns ("he", "she", "I", "you", "it") and the definite article ("the") can likewise be used to refer. In particular, the definite article is applied in the so-called definite descriptions, i.e., sentences of the form "the x such that ϕx" such as "the man who corrupted Hadleyburg". Finally, reference can be done via proper names (e.g., "Venice", "Napoleon", "John").

According to Kripke, a proper name is a *rigid designator*, i.e., "in every possible world it designates the same object" (Kripke, 1980). Nevertheless, not everything that is used to refer is a rigid designator. In Kripke's example, the definite description "the President of the U.S. in 1970" designates a certain man, Nixon. But someone else might have been the President in 1970 (e.g., Humphrey), so this designator is not rigid. Contrariwise, the proper name "Nixon" is a rigid designator. As Kripke states, "although the man (Nixon) might not have been the President, it is not the case that he might not have been Nixon (though he might not have been called 'Nixon')". Concerning the *origin* of a proper name, Kripke provides the notion of a *baptism ceremony*:

[12] http://plato.stanford.edu/entries/reference

> An initial “baptism” takes place. Here the object may be named by ostension, or the reference of the name may be fixed by a description. When the name is “passed from link to link”, the receiver of the name must, I think, intend when he learns it to use it with the same reference as the man from whom he heard it. (Kripke, 1980)

This could be related to the notions of connotation and denotation provided by (Mill, 1882). When applying these notions to proper names, Mill states (parts have been highlighted):

> Proper names are not connotative: they denote the individuals who are called by them; but **they do not indicate or imply any attributes as belonging to those individuals**. When we name a child by the name Paul, or a dog by the name Caesar, these names are **simply marks used to enable those individuals to be made subjects of discourse**. (...) Proper names are attached to the objects themselves, and are not dependent on the continuance of any attribute of the object. (Mill, 1882)

Besides proper names, we are interested in other designators that refer to individuals in reality, e.g., national identification number, social security number, GS1 country code, MAC address, postal code. Henceforth, those designators, along with proper names, are called *identifiers*. According to our characterization of the information level, identifiers are considered to be pieces of symbolic data. Further, identifiers are not exclusively related to information systems implementation, but rather may be used by humans in the social world.

We consider that the main purpose of an identifier is *denotation* (i.e., to refer to a thing), as opposed to connotation (i.e., to describe properties of a thing). Nevertheless, we do not deny that identifiers may possess connotation; in fact, they frequently do. One of the reasons for this is the convenience for humans to manipulate identifiers (specially, numbers) that connote properties of things in reality. For instance, the numbers assigned to floors in a building certainly connote height; the greater the number is, the higher the floor is from the ground. As another illustration, the format for a national identification number (used to identify people within a country) frequently includes information on date and location of birth, and sometimes on gender. However, connotation could be removed from identifiers without compromising denotation. For instance, each floor in a building could be assigned to an arbitrary symbol (e.g., a star, a leaf, a sun). In this trivial case, the only meaningful operation between two identifiers is equality. Besides that, any other operation is invalid, e.g., addition, sum, ordering.

7.1.2 Informational Decisions

Informational agents may share a domain conceptualization, but refer to individuals in a different way. For instance, a country may be referred to via a language specific name (e.g., “Brasil”), an ISO 3166 country code (e.g., “BR”), a GS1 country code (e.g., 789) and so forth. Those are simply agent-

specific ways of referring to (presumably) the same country individual. In the following, we identify informational decisions on reference, by discussing what sorts of things can be referred to by an identifier.

First, identifiers can be attributed to substantials, e.g., social security numbers for US citizens, “Allies” and “Axis” for the (non-extensional) rival groups of the World War II, numbers for blood samples in a laboratory, numbers for bank accounts. In this case, they refer to instances of Kinds.

Furthermore, identifiers can also be attributed to relator individuals (or to qua individuals that compose them). For example, a student enrollment number is an identifier specifically created to refer to the many attributes of a person while playing the role of student in the context of an enrollment. If John studies in two universities at the same time and he has two different student enrollment numbers for each, then his particular properties in each university can be tracked by each identifier. For instance, if John is enrolled in Stanford and in Harvard, one identifier may refer to “John qua student of Stanford” (to keep track of his grades and class attendance records in Stanford) and the other to “John qua student of Harvard” (to keep track of his grades and class attendance records in Harvard). Ergo, we say identifiers may refer to instances of Relators (alternatively, to qua individuals inhering in instances of Roles). Other examples include flight number, passenger id, medical exam number and driver’s license number.

The same type of identifier can be bound to individuals with different principles of identity. For example, items in a pet store that have different principles of identity (e.g., cats, cages and bird feed) may share the same type of identifier. That is to say, the type of identifier may have a correspondence not to the Kind that the individuals instantiate (e.g., Person, Organization), but rather to the Category common to all individuals (e.g., Legal Entity).

As a conclusion, in an information model, reference schemes may be applied to types corresponding to Kinds, Relators and Categories (henceforth, we call those “referable universals”). In a domain ontology, for each universal pertaining to one of those categories, it is an informational decision whether to apply a reference scheme to its corresponding type at the information level. Informational decisions may also concern the structure of each identifier type (defining symbols, syntax, check digits, etc.) in order to address certain pragmatic issues of information manipulation.

7.1.3 Model-driven Approach

In information models, we represent identifiers as attributes of types. For every “referable universal” in the domain ontology (viz. Kind, Relator, Category), one may choose to attach a reference attribute (identifier) to the corresponding type (viz. natural type, relator type, mixin type). Nevertheless, one could also rely on the default identification mechanism provided by object-orientation and, thus, could decide not to have any reference attribute attached to the type. We depict the patterns for

informational decisions on reference in Figure 7.1. On the left side, we present four universals belonging to the running example (Figure 2.8) and, on the right side, their corresponding types. Moreover, we attach a reference attribute to each type.

The reference attribute is called “id”, if not otherwise specified. Then, there are decisions concerning its data type. The attribute may be of a built-in primitive type such as string and integer (e.g., see the Employment and Legal Entity types). Additionally, the identifier may be of a user-defined data type (e.g., see the Person and Organization types). In this case, an additional data type is created in the information model (e.g., see the OrganizationID and SSN data types). If not otherwise specified, this data type bears the name of the type owning the reference attribute, plus the suffix “ID” (e.g., “OrganizationID”). In the example, for the Person type, both the reference attribute and the attribute’s data type have a custom name (respectively, “ssn” and “SSN”, standing for social security number).

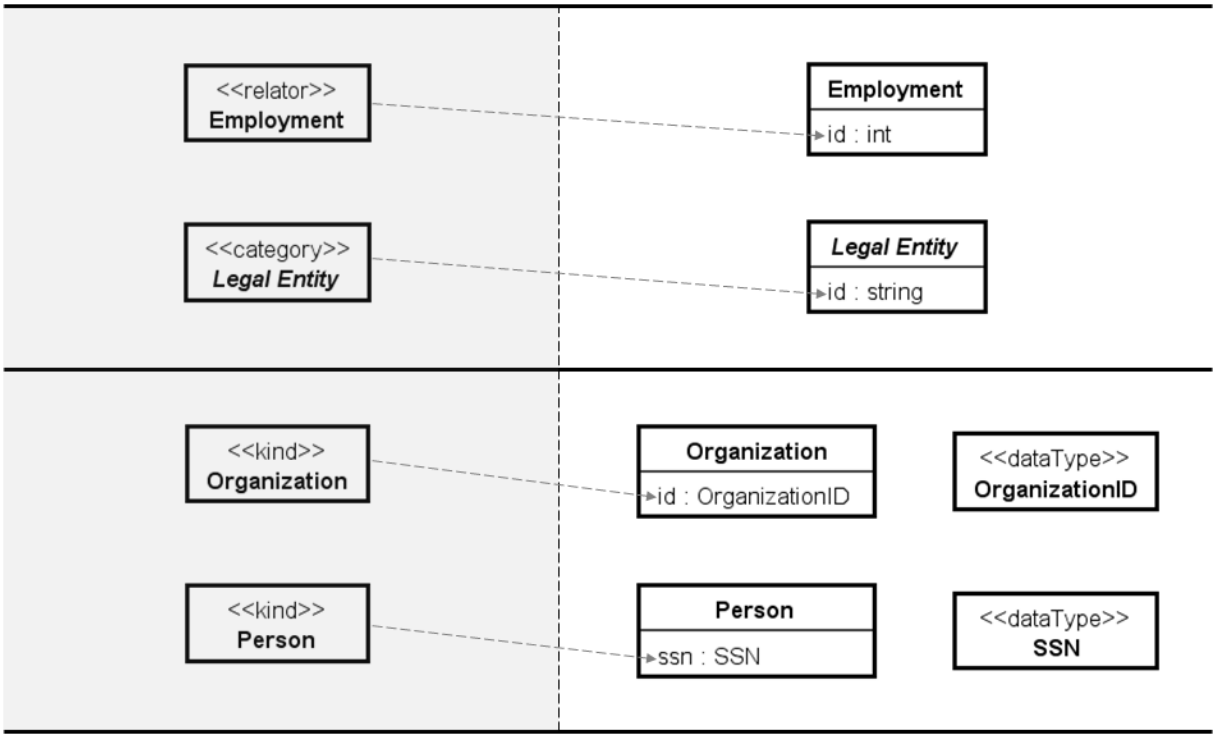

Figure 7.1 - Patterns for reference decisions (via attributes and data types)

7.2 MEASUREMENT

7.2.1 INTRODUCTION

At the ontological level, as presented in chapter 2, qualities are “the basic entities we can perceive or measure” (Gangemi et al., 2002) and are distinguished from their qualia. A quale describes the position of an individual quality within a certain quality structure. Each quality structure is “endowed with certain geometrical structures” and is supposed “to satisfy certain structural constraints”

(Gärdenfors, 2000). At the information level, measurement concerns arise from the fact that information on qualities may be obtained in different ways with different instruments according to particular information demands.

We consider that data on a quale value of a quality is obtained by means of a *measurement event*, with the assistance of a measuring instrument. Measurement events have qualitative aspects such as its accuracy, precision and granularity. Accuracy (i.e., being near to the true value) and precision (i.e., being reproducible) reflect our incapability to perfectly grasp reality through instruments. For instance, the accuracy of data on a quale value is just as accurate as the measuring instrument can be, along with other circumstances of the measurement event. Granularity (i.e., the level of detail) may reflect ignorance about further details of a certain quality or a limitation that corresponds to an information demand. For example, a coarse-grained granularity for weight may be the result of the instrument's granularity, or an information demand of having values with coarse detail.

In this thesis, we argue that identifiers should be discerned from measurement values. First of all, identifiers are used for *denotation*, while measurement values are used for *connotation*. Second, identifiers are *attributed* in a baptism ceremony, while measurement values are *measured* in a measurement event (susceptible to flaws). Third, since qualities of a thing *change*, the obtained value may be different every time a measurement event is performed. Contrariwise, since an identifier is a rigid designator that is attributed to an individual, its "value" is *constant* throughout the lifecycle of that individual.

7.2.2 Informational Decisions

All informational decisions about measurement concern Qualities in the domain ontology. Foremost, at the information level, there is freedom to choose the data type representing a certain quality structure, as long as there is a correspondence between the data type and the geometrical arrangement of the quality structure. The relation between a data type specification and a quality structure was also accounted by Guizzardi: "whatever constraints should be specified for a datatype must reflect the geometry and topology of the *quality structure* underlying this data type" (Guizzardi, 2005). As an illustration, consider the Weight Quality being related to a quality structure isomorphic to the half-line of nonnegative numbers. Then, weight values might be encoded, e.g., in real numbers (e.g., for experimental physics), in integer numbers (e.g., for high school gym classes) and in lexical spaces (e.g., "thin", "average", "overweight" in a survey).

In addition, the measurement of Qualities is related to other informational concerns, namely, scope, history tracking and time tracking. Thus, there are also informational decisions targeting

Qualities with respect to those concerns. We elaborate on them as we explain our model-driven approach for measurement.

7.2.3 Model-driven Approach

We represent data on qualia (values) of a Quality as attributes. In the domain ontology, each Quality characterizes a certain universal. Correspondingly, in the information model, a measurement attribute is possessed by the type corresponding to the characterized universal (henceforth, "characterized type"). This is illustrated in Figure 7.2. By default, the measurement attribute bears the same name of the corresponding Quality, but in lower case (e.g., "height", "weight" and "color"). Furthermore, a data type must be established for the measurement attribute, which may either be a primitive type (e.g., integer, float, string) or a user-defined data type. In the example, the Weight Quality is represented in the integer primitive type, while the Height Quality is encoded in a data type standing for Meters. A data type may be structured (i.e., may possess attributes) as in the Color example. Finally, in a characterization relation, the cardinality constraints in the Quality association end are reflected in cardinality constraints in the corresponding measurement attribute. For instance, those constraints have been highlighted in the domain ontology and in the information model for Color.

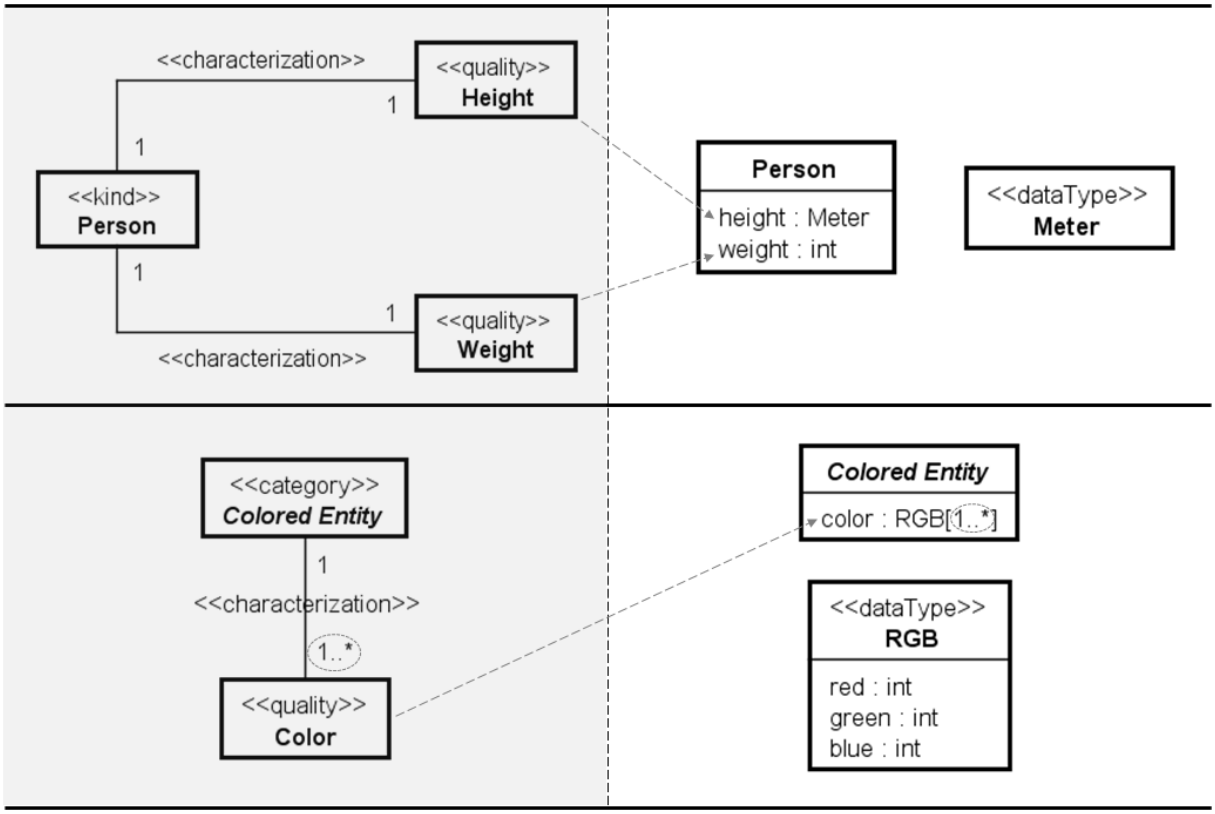

Figure 7.2 - Patterns for measurement decisions (via attributes and data types)

Qualities are also target of scope decisions, as the other meta-categories presented in chapter 5. If a Quality is outside the scope, the corresponding measurement attribute will be absent in the information model. An example is depicted in Figure 7.3, in which the Height Quality is considered outside the scope.

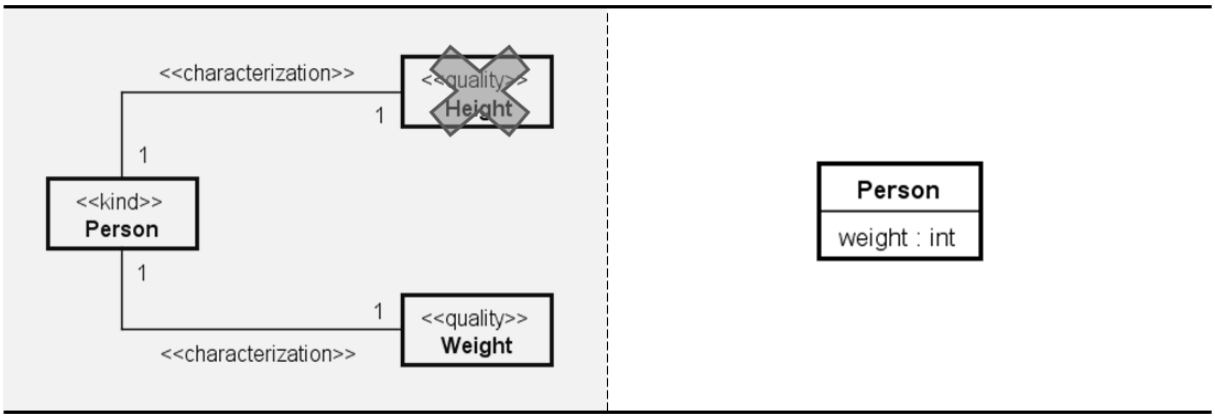

Figure 7.3 - Pattern for scope decisions on Qualities

Qualities are likewise target of history tracking decisions, similar to the meta-categories presented in chapter 6. In our model-driven approach, we must address an information demand to keep the history of measured values for a certain Quality. In this case, we represent in the information model a type whose instances are objects (data fragments) about the measurement event of the Quality (henceforth, “measure type”). We illustrate this situation in Figure 7.4. The measure types “Height Measure” and “Weight Measure” in the information model correspond, respectively, to the Height and the Weight Qualities in the domain ontology. In this approach, the measurement attribute is actually possessed by the measure type, instead of the characterized type. Then, an instance of the characterized type (e.g., Person type) is related to various instances of the measure type (e.g., Height Measure type).

Instances of the measure type must be distinguished with respect to time, especially to identify the object representing the last measurement event (storing data about the current quale value). A possible solution is to include an attribute in the measure type representing the time instant in which the measurement event occurred; this is equivalent to performing time tracking on the measurement event. We illustrated this approach in the Weight Measure type. Another solution is to rely on the ordering relation among the instances of the measure type. For example, objects may be ordered decreasingly with respect to time, being the first object consequently the representative for the last measurement (storing the current quale value). This approach is illustrated in the Height Measure type.

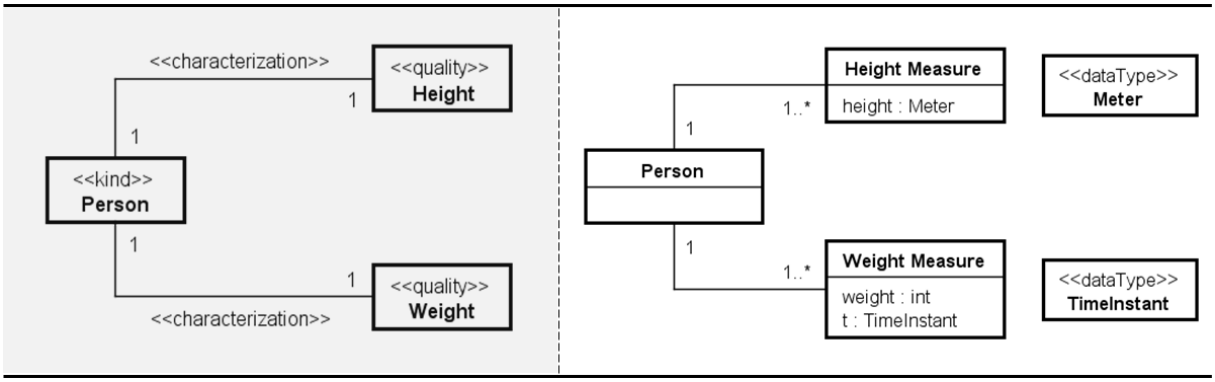

Figure 7.4 - Patterns for history and time tracking on measurement of Qualities (via measure type)

Finally, to address situations in which measurement attributes may be unknown, it is possible to add optional cardinalities in the information model. We illustrate this point in Figure 7.5, in which history tracking is required only for the Weight Quality. In the information model, the height attribute of the Person type is marked with a lower bound zero. Moreover, the relation between the Person type and the Weight Measure type also has lower bound zero.

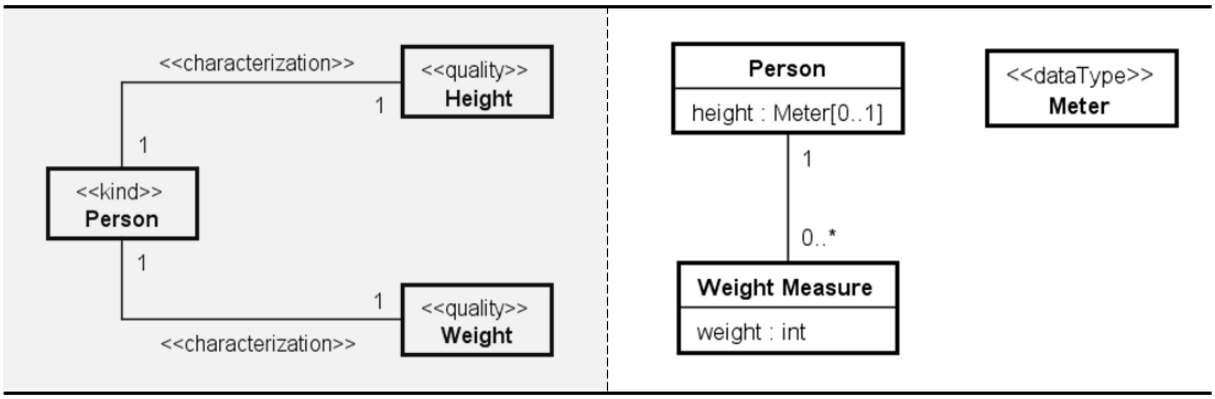

Figure 7.5 - Introduction of optional cardinalities on measurement attributes

7.3 CONCLUSIONS

In our model-driven approach, both reference and measurement are addressed via attributes and data types in the information model. Nevertheless, as we argued, the ontological interpretations underlying both sorts of attribute are distinct. We supported our argumentation on this subject on the basis of a baptism ceremony for reference and a measurement event for measurement. On the one hand, an identifier is attributed to an individual and then “passed from link to link” with denotation purposes. On the other hand, the value of a quality is measured (thus, susceptible to flaws) and has connotation purposes. The distinction between reference and measurement was also emphasized by describing informational decisions on both concerns. In depth, since identifiers are constant throughout the lifecycle of an individual, there is no notion of history tracking for reference attributes. Contrariwise, since quale values of a quality may change throughout life, there are history

tracking patterns for measurement attributes. Further, reference decisions target “referable universals” (viz. Kinds, Relators and Categories), while measurement decisions target Qualities.

Finally, there are some informational decisions that we do not systematically address here, but are worth being mentioned. There may be cases where one needs to know about specific details of a baptism ceremony or a measurement event. As an illustration for reference, one may need to know the issuing date of national identification numbers (i.e., the baptism ceremony date) and which governmental organization issued them (i.e., the baptizer). In those cases, it is helpful to explicitly represent in the information model data structures on the baptism ceremony or the measurement event.

For measurement, this may include qualitative attributes of the measurement event (e.g., accuracy, precision and granularity) and information on its participants (e.g., the measuring instrument, the measurer). Information on the measurement event corresponds to the so-called “meta-attributes” in the data quality literature (Wand & Wang, 1996). For instance, at the information level, there may be different information demands concerning weight values. In the context of experimental physics, one may be interested in various sorts of information about weight measurement. For a certain weight value, one may be interested in which physicist performed the measurement and which weight scale was used, along with experimental uncertainties (e.g., “John obtained a value of 60,000±0,003 kg for Mary’s weight using the weight scale XYZ123”). On the other hand, in the context of gym classes in a high school, one may simply need to record weight values without knowing any further detail about the measurement event (e.g., “Mary’s weight is 60 kg”).

8 TOOL SUPPORT

In this chapter, we describe the tool support for our model-driven approach, which provides a transformation from OntoUML to UML. Our tool support is implemented as a plug-in for the Eclipse platform using the Eclipse Modeling Framework (EMF), which we briefly explain in section 8.1. Furthermore, our tool explores the facilities of the so-called OntoUML Infrastructure, provided in (Carraretto, 2010), which is explained in section 8.2. We also present in this chapter fragments of the UML metamodel (section 8.3) and the OntoUML metamodel (section 8.4) that were involved in our model transformation. Finally, in section 8.5, we present our plug-in in the perspective of end-users, by showing the input, the user interface and the output.

8.1 THE ECLIPSE MODELING FRAMEWORK

The Eclipse Modeling Framework (EMF) is a metamodeling framework integrated in the Eclipse IDE that is centered on the specification of (meta)models written in the Ecore language. An Ecore (meta)model may generate, among other things, Java implementation code. Most importantly, the model code can be used to guide the implementation of *model manipulations*, e.g., model editing, persistence in different formats, interchange between conceptual modeling tools, syntactic and semantic validation, generation of textual documentation, analysis of model content and model transformations (Carraretto, 2010). Ultimately, EMF provides a key support for the implementation of tools based on the model-driven architecture (MDA). More information on EMF is provided in (Steinberg, Budinsky, Paternostro, & Merks, 2008).

8.2 THE ONTOUML INFRASTRUCTURE

The OntoUML Infrastructure (Carraretto, 2010) is a well-established modeling infrastructure used to support the development of several model manipulation tools involving the OntoUML language. The infrastructure is centered in the so-called *reference metamodel*, which is a specification of the abstract syntax of the OntoUML language written in Ecore. The reference metamodel uses a fully-compliant UML 2.0 metamodel implementation as a foundation and introduces OntoUML constructs according to the rules for forming UML profiles in the so-called *lightweight extension*. The metamodel also embeds the specification of all OntoUML syntactical constraints, written in the Object Constraint Language (OCL) (OMG, 2006). Furthermore, the reference metamodel is free from implementation commitments with respect to concrete syntax and model editing tools. Ergo, the reference metamodel is supposed to be the common ground for all OntoUML model manipulations.

One of the premises underlying the reference metamodel is the separation between graphical editing and other model manipulations. Developers of front-end graphical editing tools usually have to create their own simplified and adapted versions of the official OntoUML metamodel. Those front-end metamodels usually contain partial or adapted UML constructs, partial or adapted OntoUML constraints and auxiliary constructs for supporting tool implementation. The methodology supported by the OntoUML infrastructure is that model manipulations (besides model editing) should not be done in terms of those front-end metamodels. Rather, those front-end metamodels should be transformed to the back-end reference metamodel and then all model manipulations should be done in terms of the latter metamodel. In this approach, front-end tool developers can take the benefits already provided by the reference metamodel (e.g., syntax validation and model transformations).

The infrastructure is illustrated in Figure 8.1. The left side of the figure illustrates two OntoUML graphical editors and their respective metamodels; we call them here the AB Editor (Benevides, 2010) and the RC Editor (Carraretto, 2010). Both front-end metamodels are connected to the reference metamodel via transformations T_1 and T_2 provided in (Carraretto, 2010). The right side of the figure illustrates several model transformations from OntoUML to other target languages. Those transformation are: OntoUML to Alloy[13] (Benevides, 2010), OntoUML to SBVR (OMG, 2008), OntoUML to OWL (Zamborlini, 2011) and, the transformation proposed in this thesis, OntoUML to UML or Onto2Info.

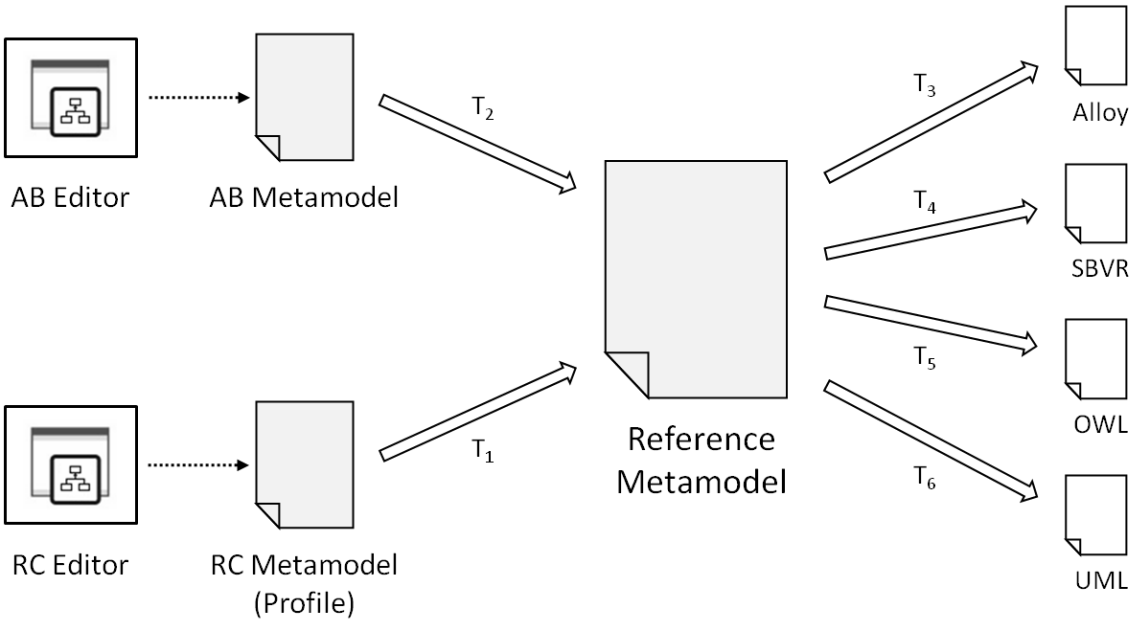

Figure 8.1 - The OntoUML Infrastructure

[13] The OntoUML to Alloy was originally implemented in terms of the AB Metamodel in (Benevides, 2010), before the existence of the OntoUML infrastructure. The transformation T_3 was a later reimplementation.

The core components of the infrastructure are built in the Eclipse environment with the support of EMF. The reference metamodel is written in the Ecore language. The two aforementioned editors are implemented in Eclipse as well. The AB Editor uses the Graphical Modeling Framework (GMF) and the RC Editor uses UML2 Tools. All transformations shown in Figure 8.1 were implemented in Java with the support of EMF. Nonetheless, the infrastructure is not limited to the Eclipse environment. As a matter of fact, recent and under development works concerning OntoUML have been developed outside Eclipse, but with the support of the OntoUML Infrastructure. Details on the OntoUML infrastructure can be found in the infrastructure's website[14].

The Onto2Info transformation requires knowledge of the UML metamodel (shared by the source OntoUML models and the target UML models) and the OntoUML metamodel (for the source OntoUML models). We present relevant fragments of those metamodels (written in Ecore) in the following sections.

[14] http://code.google.com/p/rcarraretto/

8.3 The UML Metamodel

For the Onto2Info implementation no effort was required to implement the UML metamodel, as we used an Ecore implementation of UML2 provided by the Eclipse Model Development Tools (MDT) project. Here, we are only concerned with some fragments of the UML class diagram, depicted in Figure 8.2, Figure 8.3 and Figure 8.4. Those fragments are not only relevant to UML information models but also to OntoUML domain ontologies.

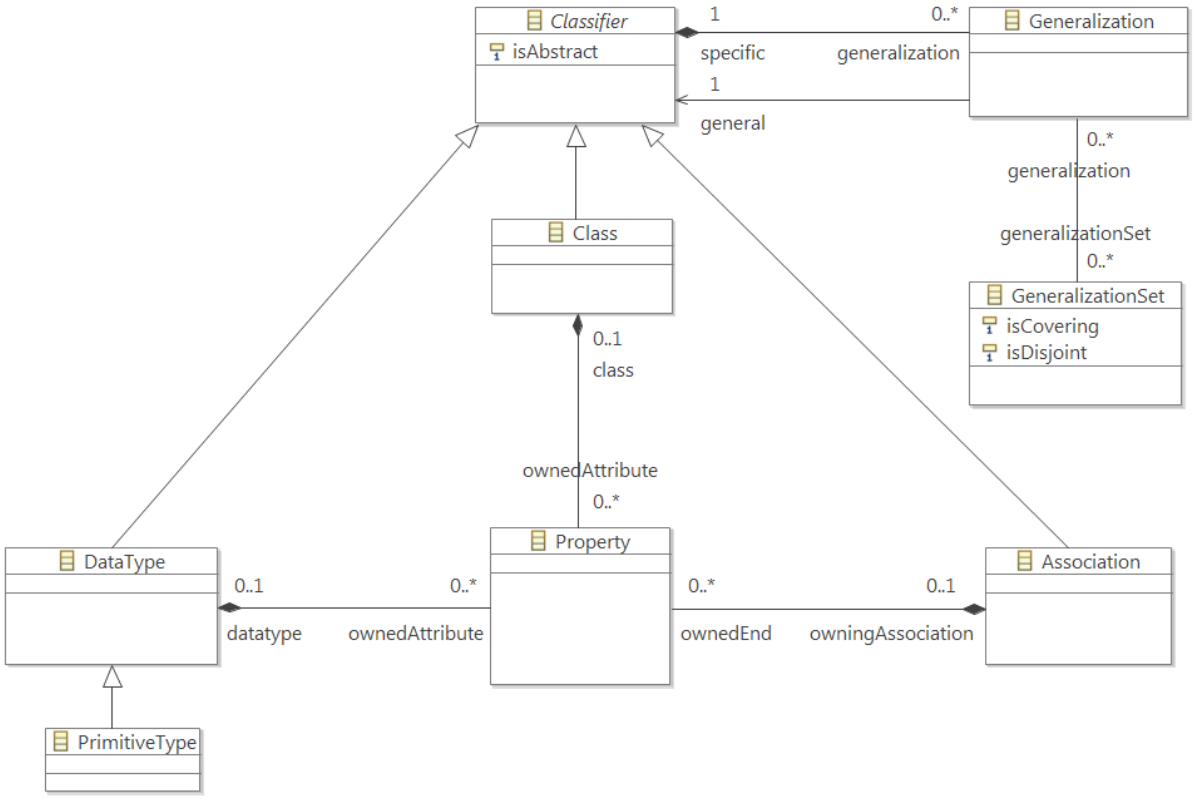

Figure 8.2 - Fragment of the UML metamodel concerning Classifier

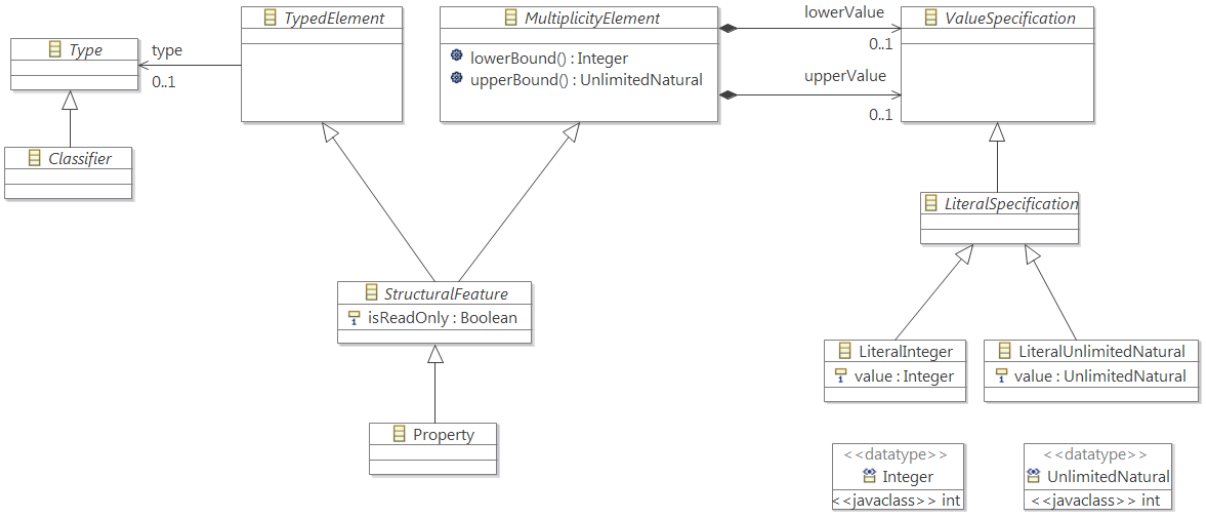

Figure 8.3 - Fragment of the UML metamodel concerning Property

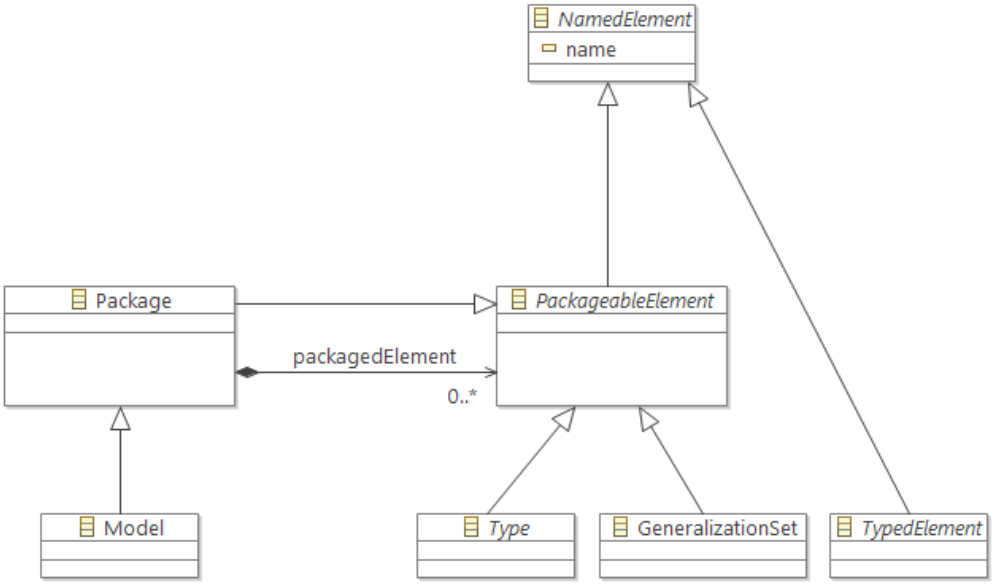

Figure 8.4 - Fragment of the UML metamodel concerning Package

8.4 The OntoUML Metamodel

For OntoUML domain ontologies, we are concerned with two fragments of the reference metamodel, depicted in Figure 8.5 and Figure 8.6.

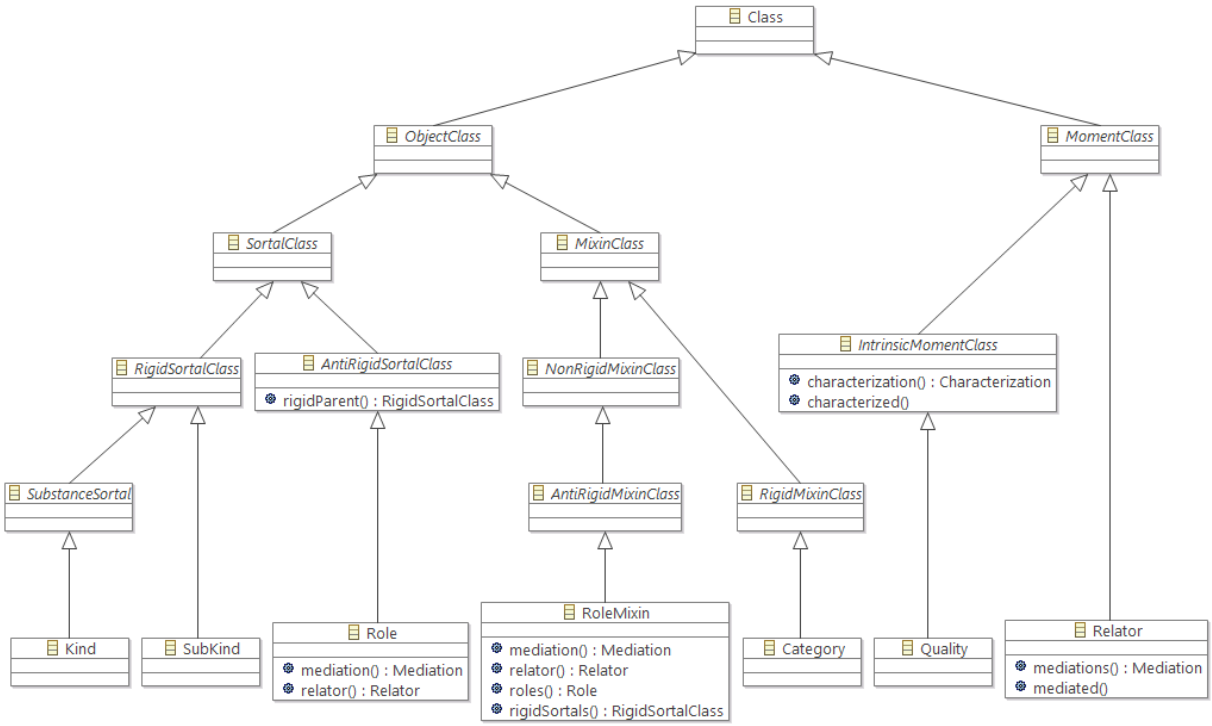

Figure 8.5 - Fragment of the OntoUML metamodel concerning the stereotypes of the UML Class

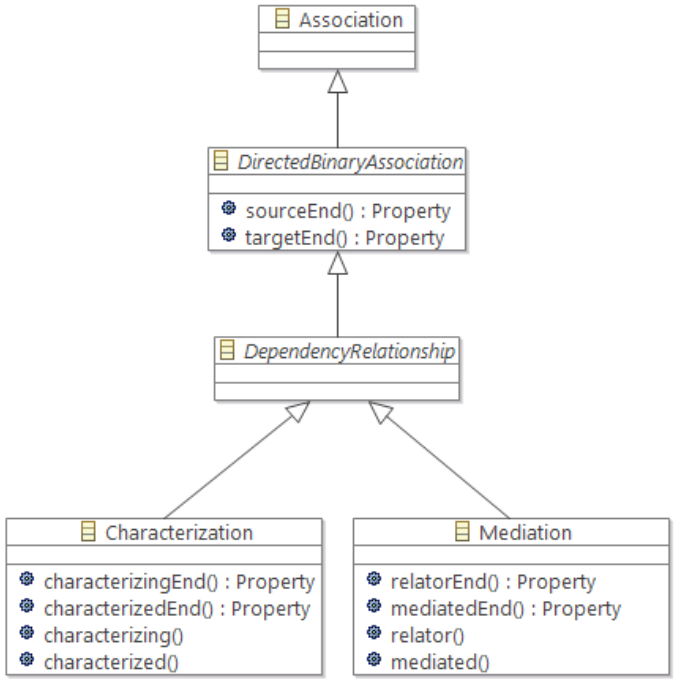

Figure 8.6 - Fragment of the OntoUML metamodel concerning the stereotypes of the UML Association

8.5 THE ONTO2INFO PLUG-IN

In this section, we present the Eclipse plug-in developed in the context of this thesis, shortly known as "Onto2Info"[15]. In Figure 8.7, we depict the installation process of the Onto2Info plug-in in Eclipse.

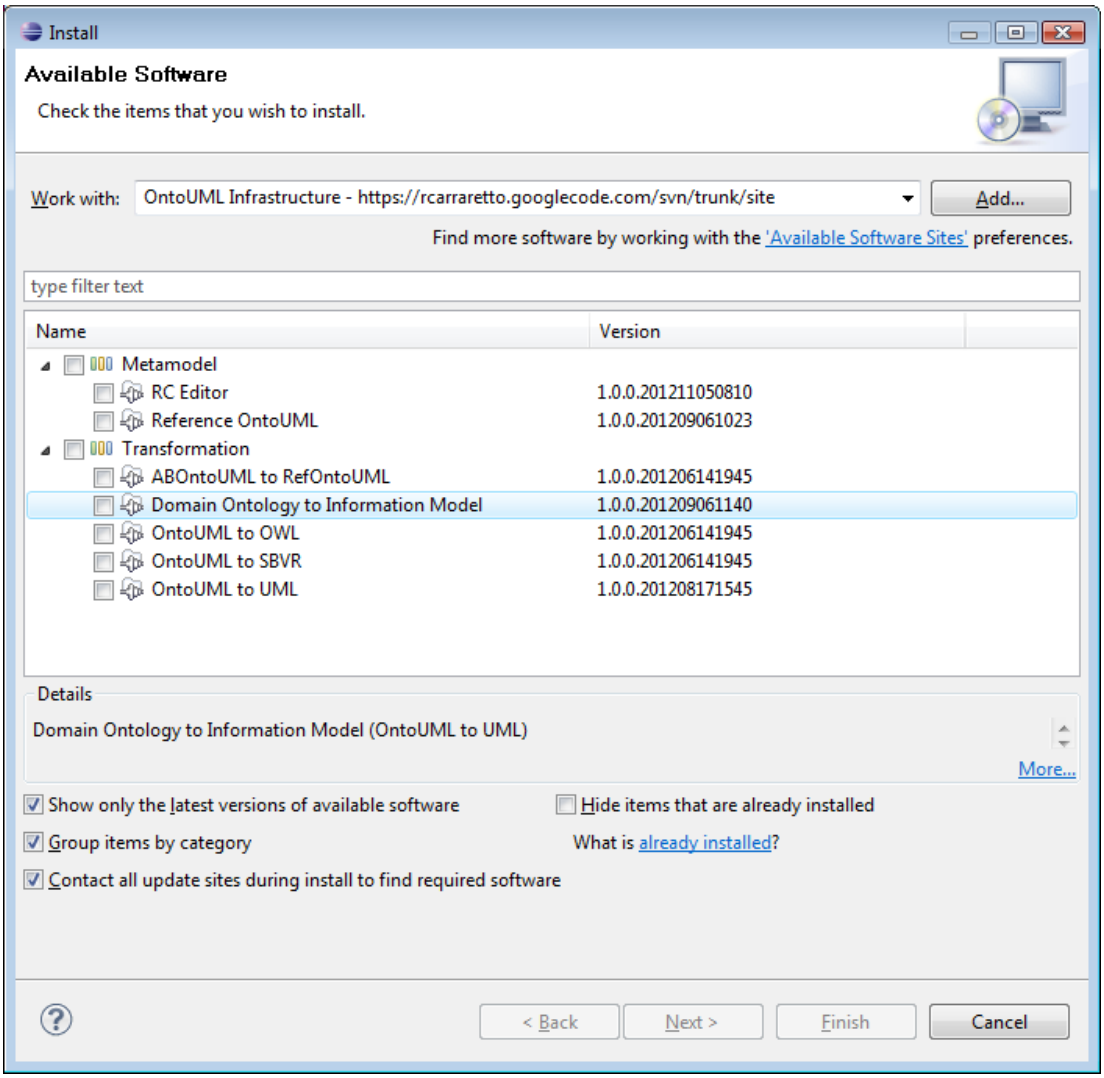

Figure 8.7 - The Onto2Info plug-in along with other plug-ins of the OntoUML Infrastructure

The Onto2Info plug-in takes as an input an OntoUML model file, written in Reference OntoUML (".refontouml" extension). The plug-in is loaded by right-clicking an OntoUML model in the workspace and by selecting the "Transform to Information Model" option in the context menu. This is illustrated in Figure 8.8.

[15] We avoid the usage of the term "OntoUML2UML" as the OntoUML Infrastructure already supplies a plug-in with this name, related to a different transformation (one that simply removes the stereotypes from an OntoUML model, transforming it in a traditional UML model).

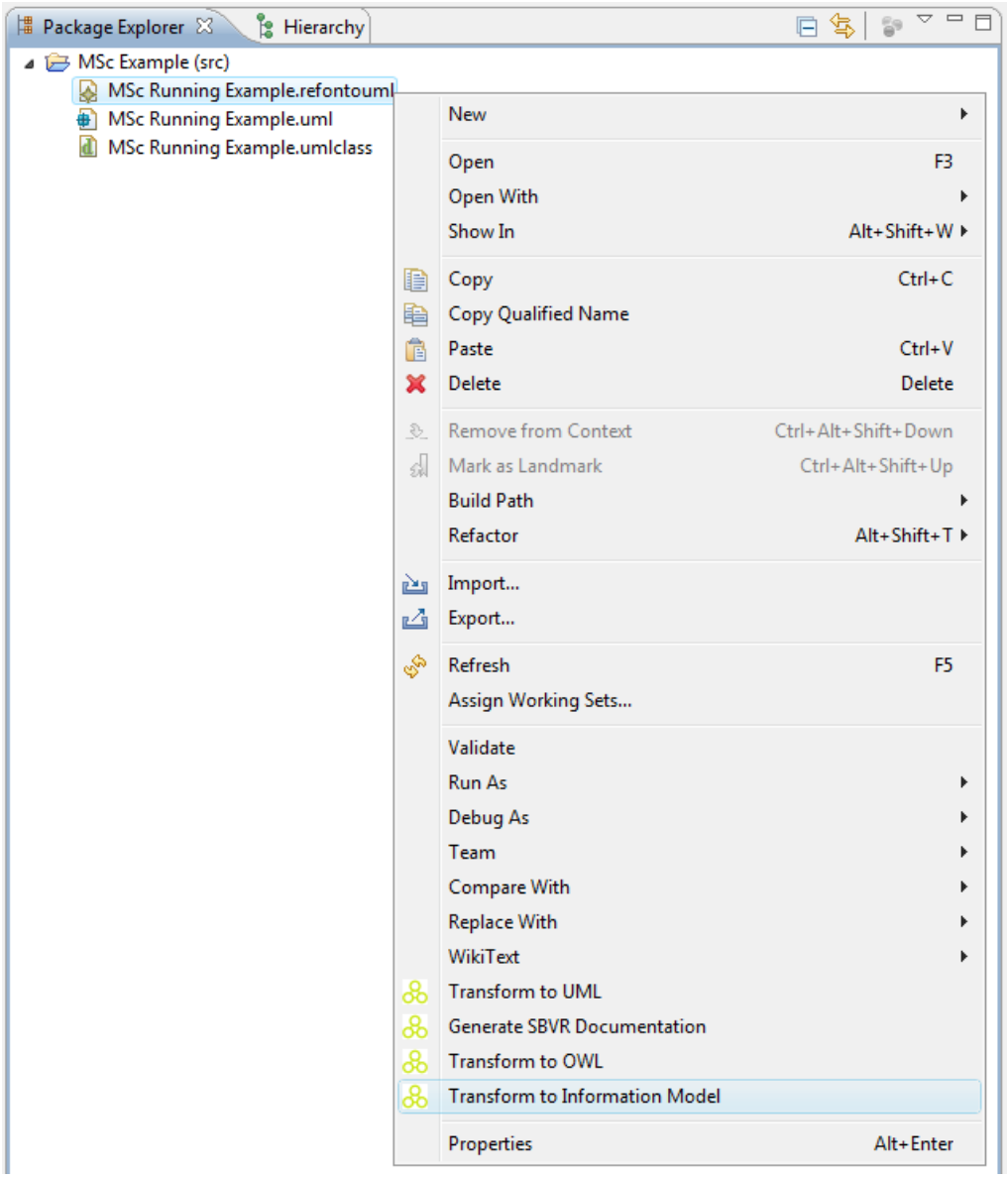

Figure 8.8 - Activating the Onto2Info plug-in via the context menu

After that, the Onto2Info interface is shown (Figure 8.9). There, the user can input the several informational decisions that precede the model transformation. The interface is divided in five tabs corresponding to the five informational concerns addressed here. We explain each tab in the following.

In the scope tab, the user decides which universals in the OntoUML model are inside or outside the scope via checkboxes (Figure 8.9). All universals in the model are shown, except Qualities – we address those in the measurement tab. Since we considered that scope decisions concerning Kinds and SubKinds affect scope decisions of specializing universals, we organize our interface in a hierarchical (tree) structure. Kinds are presented as top nodes of the hierarchy and subsume specializing SubKinds and Roles. If a node in a hierarchy is unchecked, all the children nodes become unchecked as well. For example, if the Person node is unchecked, then all children nodes are likewise

unchecked (viz. Man, Woman, Husband, Wife, Pregnant, Fetus, Employee and Private Customer). In addition, if a node in the hierarchy is checked, all the parent nodes become checked as well. For instance, if the Wife node is checked, the Woman and the Person nodes are then checked. After hierarchies, we display Categories, Role Mixins and Relators.

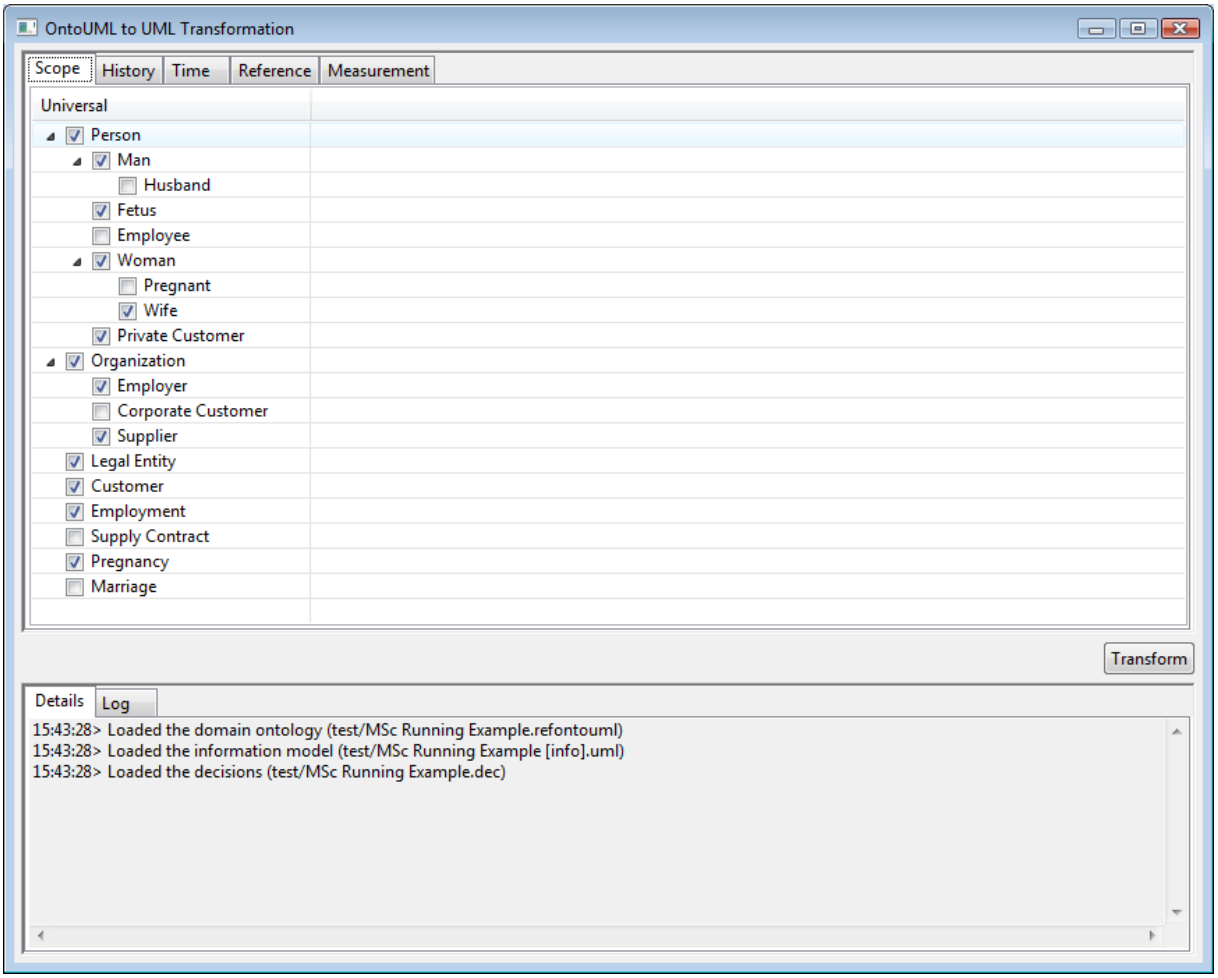

Figure 8.9 - The Onto2Info interface, showing the scope tab

In the history tracking tab, for each universal, the user manifests interest on past and/or present via two checkboxes (Figure 8.10). Only Kinds and Relators are displayed (Qualities are treated in the measurement tab).

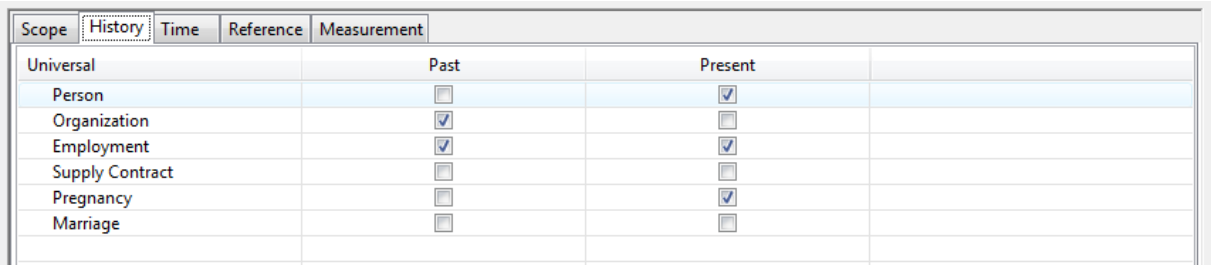

Figure 8.10 - The history tracking tab

Analogously, in the time tracking tab, for each universal, the user manifests interest on start time, end time and duration via checkboxes (Figure 8.11). Once more, only Kinds and Relators are shown (Qualities are treated in the measurement tab).

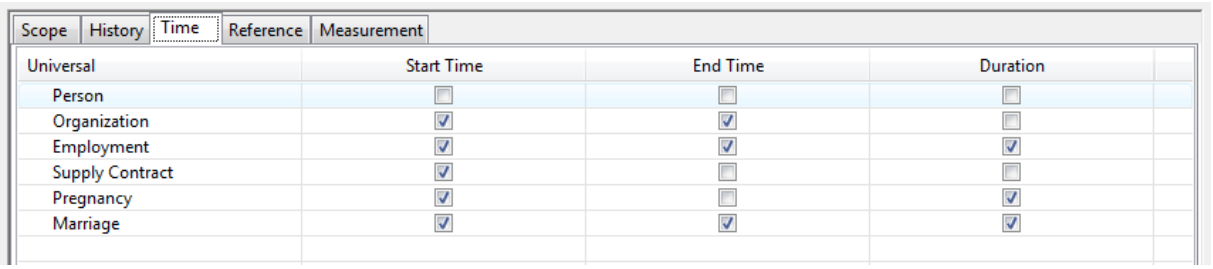

Figure 8.11 - The time tracking tab

In the reference tab, for each universal, the user decides via checkbox whether a reference attribute should be attached to the corresponding type (Figure 8.12). Moreover, the user may edit via textbox the name of the reference attribute (originally called “id”). Via drop-down list, the user can select whether the attribute is of a primitive type (e.g., integer, string) or a custom data type. In the latter case, the user may alter the default data type name (e.g., “EmploymentID”), if desired.

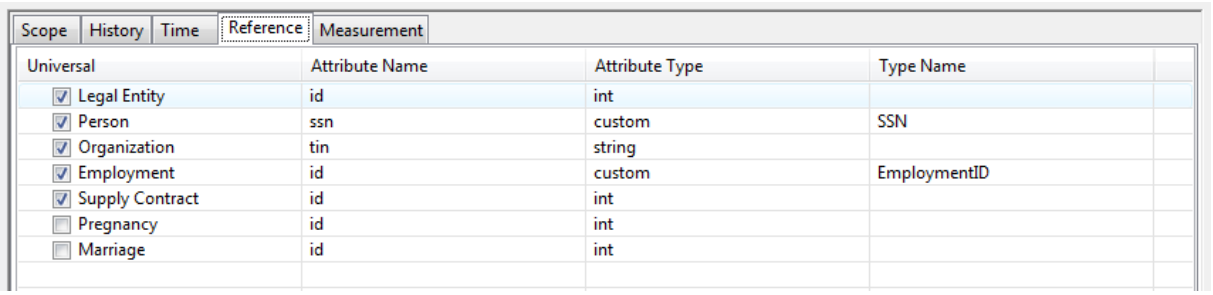

Figure 8.12 - The reference tab

Finally, in the measurement tab, for each Quality universal, the user may decide via checkbox whether the Quality is inside or outside the scope (Figure 8.13). We display the characterized universal of each Quality in a read-only column. Similarly to the reference tab, the user may select via drop-down list whether the measurement attribute is of a primitive or a custom type. In the latter case, the user may change the default type name (e.g., “HeightValue”), if desired. Furthermore, interest on history and time tracking is manifested in two checkboxes. The time checkbox may only be checked if the history checkbox is checked.

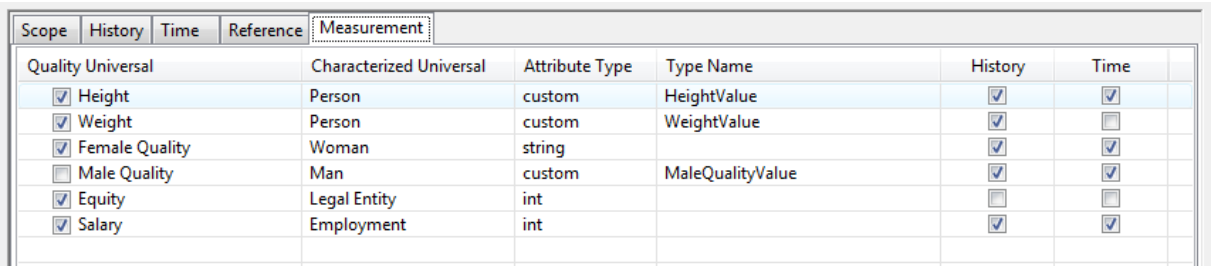

Figure 8.13 - The measurement tab

Once the user has performed the informational decisions, the output UML model can be generated via the "Transform" button in the interface. The output model has the ".uml" extension and can be opened in Eclipse Modeling, as illustrated in Figure 8.14. The ".uml" file only contains abstract syntax elements. Nonetheless, in Eclipse Modeling, one can generate an initial class diagram for a ".uml" file via the "Initialize Class Diagram" option in the context menu. Then, a ".umlclass" file containing concrete syntax elements is created.

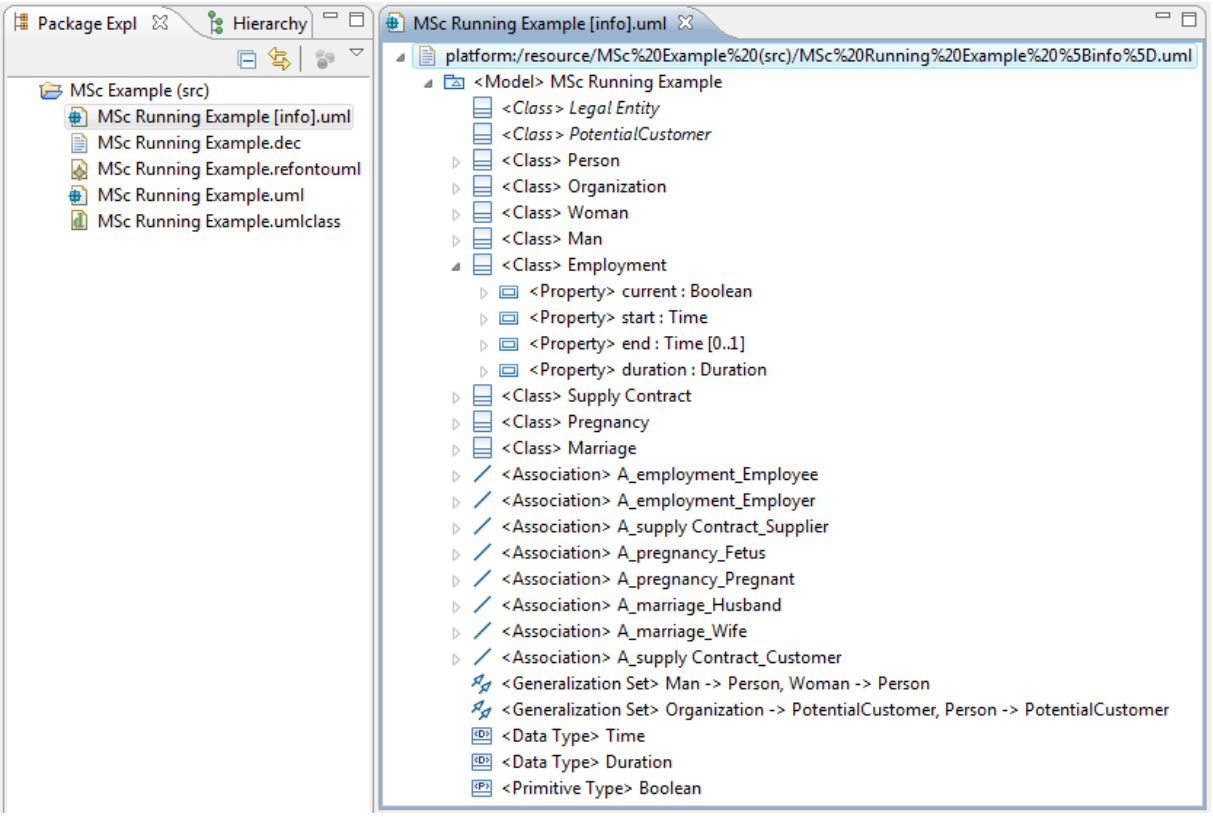

Figure 8.14 - An output UML model generated by the Onto2Info plug-in

It is worth to notice that our output UML model receives the same name as the input OntoUML model, but with an "[info]" suffix attached to it. We explain this issue in the following. OntoUML models can be created via UML profile-based graphical tools, such as the RC Editor provided in the OntoUML Infrastructure. Then, an OntoUML model may be stored in two files, one with a ".uml"

extension (for the abstract syntax) and other with a ".umlclass" extension (for the concrete syntax). This is illustrated in Figure 8.14, in the files "MSc Running Example.uml" and "MSc Running Example.umlclass". Those, in fact, represent an OntoUML model. The ".uml" file was later converted to the "MSc Running Example.refontouml" file (written in the Reference OntoUML metamodel). Thus, in order to distinguish OntoUML models written in UML profile-based tools (as domain ontologies) from our output UML models (as information models), we use the "[info]" suffix.

The Onto2Info plug-in also generates an extra output file with ".dec" extension, namely, the decision file. This file stores the informational decisions that were taken by the user in order to generate the UML information model. With this file, it is possible to visualize the informational decisions in the user interface, even after the interface has been closed. Thus, the user may later reopen the Onto2Info interface and reassess the informational decisions.

Another important feature of the decision file is that it stores the correspondence relations between constructs in the OntoUML model and constructs in the UML information model. For example, the decision file not only stores that the Person Kind is inside the scope, but it also stores that the Kind named "Person" in the ".refontouml" file corresponds to the class named "Person" in the ".uml" file. As a consequence, constructs in the UML model may suffer minor changes (such as being renamed) and may still be kept in correspondence with constructs in the OntoUML model.

When a transformation is re-done in the Onto2Info plug-in (with different informational decisions), only the inexistent constructs are added to the UML model, i.e., if there is already a corresponding construct, nothing is changed in the UML model. So, the plug-in displays to the user the actual changes that were done in the UML information model, when compared to the previous transformation. The number of changes is displayed in the details tab (Figure 8.15) and the actual changes are displayed in the log tab (Figure 8.16).

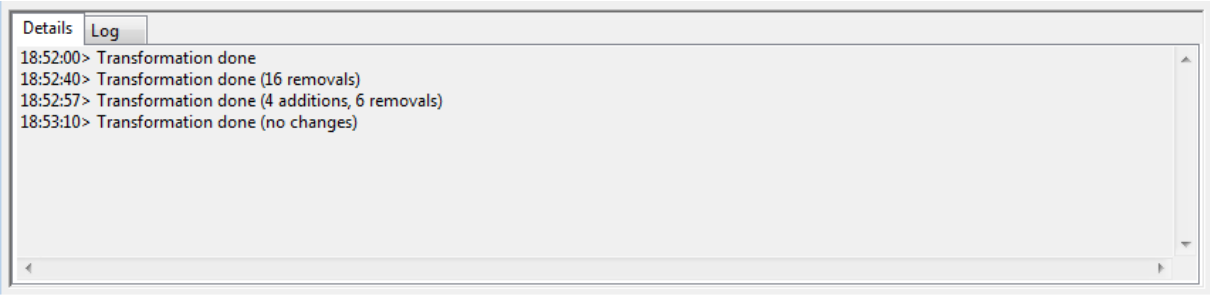

Figure 8.15 - The details tab

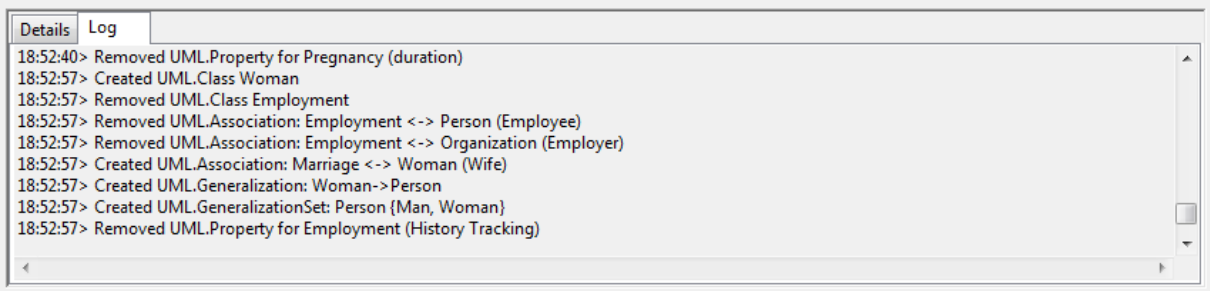

Figure 8.16 - The log tab

8.6 CONCLUSIONS

Considerable effort was made to operationalize our model-driven approach via tool implementation. We implemented the model transformation as an Eclipse plug-in incorporated in the OntoUML Infrastructure. Our transformation was implemented in Java using two Ecore metamodels: the OntoUML reference metamodel (pertaining to the OntoUML Infrastructure) and the UML metamodel (provided by Eclipse MDT-UML2). Concise fragments of both metamodels were presented here. We have shown that informational decisions can be systematically captured by means of a graphical user interface consisting of several tabs for each informational concern. We illustrated how informational decisions and correspondence relations between an OntoUML model and a UML information model may be kept in a decision file. In our implementation, the user may reassess his informational decisions and examine the relative impacts in the information model. Hence, our Onto2Info plug-in demonstrates the feasibility of our model-driven approach and also adds value to the OntoUML Infrastructure.

9 Related Work

In this chapter, we provide discussions on related work. Initially, in section 9.1, we examine efforts that advocate some separation of concerns in conceptual modeling, indicating some sort of two-level approach. We discuss two early works, (Langefors, 1980) and (Ashenhurst, 1996), that are based on the distinction between "infological" and "datalogical" aspects, and we elaborate on two early works on ontology, (Gruber, 1995) and (Guarino, 1994).

Afterwards, in section 9.2, we discuss efforts that serve to illustrate how the distinction between ontology-based modeling and information modeling is not completely grasped in the literature. More specifically, we present our concerns with respect to the works of (Jarrar & Meersman, 2009) and (Halpin & Morgan, 2008). In that section, we also discuss how information (and conceptual data) modeling approaches could contribute to a more sophisticated handling of informational concerns in our information level approach, using (Halpin & Morgan, 2008) as an example. In section 9.3, we compare our model-driven approach with the one provided in (Zamborlini, 2011), which is related to OntoUML and the OWL language.

Then, in section 9.4, we investigate works on the distinction between ontology and epistemology, (Bodenreider, Smith, & Burgun, 2004) and (Atmanspacher, 2002), which partially intersect with the distinction between ontological and informational concerns addressed here and could provide some insights for further investigation. Lastly, in section 9.5, we present concluding remarks.

9.1 Efforts on Similar Separation of Concerns

9.1.1 Langefors

Langefors (Langefors, 1980) contrasts two aspects of the so-called "user views", namely, the infological aspect (the "user view of the world") and the datalogical aspect (the "user view of the data"). The former is "concerned with how conception relates to data and information, and to reality", while the latter is "concerned with the selection of data from a data base and the rearrangement of them to suit a 'user view' of the data".

At that time, early in the 80's, works on database and structured-programming had already stressed the importance of describing data independently from representational details – for instance, (Chen, 1976). Nevertheless, Langefors argued that they were still focusing on data and on "how the same data are processed distinctly in distinct applications", instead of focusing on the world. He advocated the definition of a "user view of the world" independently of the implementation of information manipulation using computers.

While Langefors has drawn attention to some important distinctions in conceptual modeling (such as those used in chapter 3 to characterize the distinction between data and information), his infological aspect (the "user view of the world") does not shed light into the distinction between ontological and informational concerns.

9.1.2 ASHENHURST

Ashenhurst (Ashenhurst, 1996) contrasts what he calls "information modeling" (or "semantic data modeling") with "data modeling". He discusses the blur between datalogical/infological aspects, e.g., the confusion "between symbolic artifacts and their interpretation as meaningful representations of, or statements about, the real world**"** (Ashenhurst, 1996). According to Ashenhurst, one source for this confusion is that constructs in a "data model" usually possess familiar labels (e.g., "Person"), which undesirably suggest that reality is being modeled, instead of data:

> (...) in either data models or information models, there is need for "labels" to distinguish record types, etc. on the one hand and entity types, etc. on the other. The use of "meaningful" labels for record types and fields in the data model context inevitably leads to verbal reference to them as if they were the [infological] objects and [infological] characteristics represented by them. (Ashenhurst, 1996)

In his view, this confusion is especially evident in the usage of object-oriented approaches: "the object-oriented paradigm has taken on a dual role, [namely,] that of representing realworld objects as well as that of realizing them in software" (Ashenhurst, 1996). As he later complements:

> (...) in the OO context where "objects" are software constructs having "identity", incorporating "methods", and operating subject to the rules of "inheritance" and "encapsulation", it is customary to think of the datalogical object CUSTOMER as corresponding to the infological CUSTOMER, which in turn corresponds to the actual customer. (Ashenhurst, 1996)

Ashenhurst was not satisfied with the "semantic data modeling" approaches at that time, such as the Entity-Relationship model and the object-oriented paradigm. Consequently, he proposes his own approach for "information modeling", called SCRIM:

> An information model should be concerned with two aspects of **the representation of reality**, namely **what there is** and **how it appears** (...) In SCRIM, therefore, information modeling is regarded as having two interacting realms, the **objective** and the **subjective**. (Ashenhurst, 1996)

In SCRIM, the "objective representation involves the notions of being and belonging (what exists and how related)". More specifically, by "being" he refers to ontological concerns of existence (identity, individuation, meta-categories, etc.) and by "belonging" he refers to moments and relations

(especially part-whole). Ergo, the objective representation is focused on ontological concerns ("what there is").

Moreover, "subjective representation is concerned with referring and inferring (reference to what exists and how related, and inference of additional beings and belongings)". "The subjective aspect of an information system is defined in terms of how the user(s) want to interrogate, update, and manipulate the information therein, which in itself represents the objective aspect, reflecting 'that which is' in the world of the application". The subjective realm is based on "presentations" of the objective representation, consisting of "views (e.g. SQL table displays)" and "statements (e.g. SQL queries)". Much of the subjective realm seems to be composed of derived things such as counts, averages, statistical attributes and things "which are not attributes of a single object". Hence, the subjective representation seems to be focused on informational concerns ("how it appears").

Thus, Ashenhurst's "information modeling" refers to both the ontological level and the information level discussed here, corresponding to what he calls the objective realm and the subjective realm, respectively. Nevertheless, Ashenhurst's approach is not quite clear in terms of methodology and apparently was not further elaborated. It is worth to mention that we have not covered derived data/information at our information level, as described in Ashenhurst's subjective realm. However, we acknowledge derived data/information as an informational concern and leave the topic for further investigation in future works.

9.1.3 GRUBER

Gruber affirms that "we use common ontologies to describe *ontological commitments* for a set of agents so that they can communicate about a domain of discourse without necessarily operating on a globally shared theory" (Gruber, 1995). Furthermore, Gruber relates the idea of ontological commitments to the "knowledge-level" described in (Newell, 1982). According to Gruber: "the knowledge-level is a level of description of the knowledge of an agent that is independent of the symbol-level representation used internally by the agent" (Gruber, 1995). He proposes "a preliminary set of design criteria for ontologies whose purpose is knowledge sharing and interoperation among programs based on a shared conceptualization" (Gruber, 1995). Among the criteria, we stress the following:

> Minimal encoding bias: The conceptualization should be specified at the knowledge level without depending on a particular symbol-level encoding. An *encoding bias* results when representation choices are made purely for the convenience of notation or implementation. Encoding bias should be minimized, because knowledge-sharing agents may be implemented in different representation systems and styles of representation. (Gruber, 1995)

Moreover, Gruber provides a case study that "involves an ontology for sharing bibliographic data, such as the information in a library's card catalog and the reference list at the end of a scholarly paper" (Gruber, 1995). On the analysis of his solution in the ontology design, Gruber gives a relevant statement:

> At this point we have enough context to examine the rationale for the central design decision in this ontology: **to define conceptual entities described by the data, rather than just specifying the data**. Why are documents distinguished from references? Why introduce publishers and authors as independent entities when only their names appear in references? Why is a theory of time points included, when dates in references always appear as numbers and strings? Do the anticipated applications need to reason about dates, publication events, authorship, and so forth? If not, why ask them to commit to these concepts in an ontology? (Gruber, 1995)

One of the answers to those questions is the following:

> Representing the conceptual entities in addition to the data (...) provides some independence from application-specific encoding and resolution. We distinguish time points, rather than just talking about year numbers, for this reason. We say that documents are published at time points in historical time, which, when mentioned in references, may be approximated at various levels of precision. If one reference claims that a journal article was published in "1993" and another that the article was published in "March 1993", then we have the representational machinery to determine that these two references are consistent. We could, instead, insist that all dates be represented in some canonical form, such as day/month/year. However, this would be specifying precision that may not be available in the data, and would vary over implemented systems. Since this commitment would follow from a particular format rather than from the conceptualization, it would be a kind of encoding bias. (Gruber, 1995)

Ergo, the distinctions between the knowledge-level and the symbol-level are akin to those taken here between the ontological level and the information level. We are in conformance with Gruber's approach, acknowledging the importance of an accurate representation of reality at the ontological level, preceding the development of information models at the information level. Gruber's work provides insightful examples of the distinction between modeling reality and modeling data. Nonetheless, to the best of our knowledge, those distinctions were not further elaborated or applied systematically in a method.

9.1.4 GUARINO

In a seminal paper (Guarino, 1994), Guarino revisits a classification of the various primitives used by knowledge representation systems, originally provided by Brachman. Brachman argued that primitives could be grouped into four levels (namely, implementational, logical, conceptual and

linguistic) – each level corresponding to an explicit set of primitives offered to the knowledge engineer. At the logical level, primitives are propositions, predicates, logical functions and operators, which are extremely general and ontologically neutral. For instance, one may use predicates in arbitrary ways, to represent "a property of a thing, the kind the thing belongs to, a role played by the thing, among other possibilities" (e.g., both Apple and Red are admissible predicates) (Guizzardi, 2005).

To improve the "flatness" of the logical languages, Brachman proposed the introduction of an epistemological level between the logical and conceptual levels. According to (Guizzardi, 2005), the rationale behind the design of languages for the epistemological level is the following: (i) the languages should be designed to capture interrelations between pieces of knowledge that cannot be smoothly captured in logical languages; (ii) they should offer structuring mechanisms that facilitate understanding and maintenance, they should also allow for economy in representation, and have a greater computational efficiency than their logical counterparts; (iii) finally, modeling primitives in these languages should represent structural connections in our knowledge needed to justify conceptual inferences in a way that is independent of the meaning of concepts themselves. Guizzardi points as examples of epistemological level languages Brachman's KL-ONE and its derivatives (including the semantic web languages such as OWL) as well as object-based and frame-based modeling languages (such as EER and UML). Moreover, Guizzardi states that "the design of epistemological languages puts a strong emphasis on the inferential process, and the study of knowledge is limited to its form, i.e., it is '*independent of the meaning of the concepts themselves*'. Therefore, the focus of these languages is more on formal reasoning than on (formal) representation" (Guizzardi, 2005).

On top of the epistemological level, Guarino introduced the ontological level employed in this thesis. While the epistemological level is the level of structure, the ontological level is the level of meaning. Foremost, we agree with the main purposes of both levels (meaning and structure) as discussed by Guarino and Guizzardi. Since the epistemological level focus on structure, rather than meaning, it resembles the information level discussed here. While Guarino and Guizzardi have focused mostly on defending the ontological level (i.e., defending the use of ontological theories at an ontological level in conceptual modeling), we focused on informational decisions that can be treated separately, but in conformance with the ontological level. We have observed that little attention has been given in the literature to how a domain ontology should be used to derive information models. So, we address informational decisions referring to meta-categories of the ontological level.

9.2 Efforts that Blur the Concerns

9.2.1 The Dogma Approach

Jarrar and Meersman present "a methodological framework for ontology engineering" called DOGMA (Jarrar & Meersman, 2009). The DOGMA approach prescribes that one should build two separate components, a "domain axiomatization" and "application axiomatizations". Those components are defined as follows: "While a domain axiomatization focuses on the characterization of the **intended meaning (i.e. intended models) of a vocabulary** at the domain level, application axiomatizations focus on the **usability** of this vocabulary according to certain application/**usability perspectives** and specify the legal models (**a subset of the intended models**) of the application(s)' interest" (Jarrar & Meersman, 2009). As we explain in the following, we are highly against the notions provided by those authors in the DOGMA approach.

A "domain axiomatization" is captured in what they call an "ontology base", which is a set of tuples representing binary relationships between linguistic terms (e.g., for the terms "Person" and "Order" they present a tuple "<Commerce: Person, Issues, Issued by, Order>"). An "application axiomatization" is used to specify "the legal models (a subset of the intended models)" and is captured by means of a "data model". In the authors' perspective, the difference between an "ontology base" and a "data model" is merely that of scope. This is evidenced in the following statement: "(...) **application-independence** is the main **disparity** between an ontology and a classical *data schema* (e.g. EER, ORM, UML, etc.) although **each captures knowledge at the conceptual level**". Furthermore, the authors claim that "(...) if a methodology emphasizes usability perspectives, or evaluates ontologies based only on how they fulfill specific application requirements, **the resultant ontology will be similar to a conceptual data schema** (or a classical knowledge base) **containing specific** –and thus less reusable– **knowledge**".

As we argued throughout this thesis, an information model (or conceptual data model) specifies the syntax of well-formed data, as opposed to phenomena in reality. Consequently, the difference between a domain ontology and an information model is not merely one of structure, generality/independence of application, usability and the like. We conclude that the DOGMA approach does not clarify the distinction between ontological and informational concerns. This evidences the importance of the theoretical foundations provided in this thesis, as it shows that the distinctions provided here are not always properly applied in the conceptual modeling literature.

9.2.2 The ORM Language

In line with a two-level approach, (Halpin & Morgan, 2008) stresses the difference between the "real world" and the "recorded world". Conceptual schemas written in ORM are supposed to represent

the “**recorded world**” and their constraints should be interpreted “as applying to the database, not necessarily to the real world” (Halpin & Morgan, 2008). Because the “real world” and the “recorded world” share some correspondence, the constraints in ORM conceptual schemas “should be at least as strong as those that apply in the real world” (Halpin & Morgan, 2008). For instance, things that are necessary in the “real world” (e.g., ORM mandatory roles) may be specified as optional information in the “recorded world” (e.g., ORM optional roles), due to our ignorance (or scope) about the domain. Those aspects enforce that ORM has a strong bias in addressing informational concerns, as opposed to ontological ones. Nevertheless, (Halpin & Morgan, 2008) also provide discussions in terms of the real-world. As a matter of fact, Halpin refers to the ontological level approach adopted here, namely, the one provided in (Guizzardi, 2005):

> While the aforementioned research provides valuable contributions, we have some reservations about its use in industrial information systems modeling. Our experience with industrial data modelers suggests that the seven-stereotype scheme would seem overly burdensome to the majority of them. To be fair, we've also had pushback on the expressive detail of ORM (...). (Halpin & Morgan, 2008)

Simplicity is then a justification for the weak ontological ground of ORM; and the same is true for languages such as ER and UML. Consequently, most information modeling (or conceptual data modeling) languages deal with both ontological concerns (viz. determining what types of things exists in the domain) and informational concerns (viz. determining what is relevant to know about those types) in an undiscerning manner. Then, informational concerns are not addressed in terms of ontological aspects. Contrariwise, in this thesis, for each informational concern, we identified a number of informational decisions that are based on ontological categories (kinds, roles, mixins, relators, qualities, etc.). As an illustration on the subject, Halpin (Halpin & Morgan, 2008) provides modeling guidelines in ORM for history and time tracking, e.g., when one needs to maintain full or partial history of things. For that, it suggests modelers should introduce additional objects for history tracking (cf. the “WeightMeasurement” object in Figure 1.7). Since these are ordinary objects, history tracking is not explicitly separated from the things being tracked, which is the case in the two-level approach discussed here.

Nevertheless, information modeling approaches may provide several contributions to our approach at the information level. For instance, ORM deals with reference (and measurement) through the so-called “reference schemes”, addressing aspects such as compound reference, disjunctive reference, context dependent reference, preferred identifiers, rigid identifiers, information bearing identifiers, etc. (Halpin & Morgan, 2008). Such sophisticated support indicates that the variety of informational decisions involved in dealing with reference should be treated

separately from the ontological level. ORM also deals with derived attributes, which were not addressed here, but could be subject of future works.

9.3 Effort that Provides a Related Model-driven Approach

A model transformation from OntoUML to OWL was provided in (Zamborlini, 2011). Her approach is based on the distinction between a "reference ontology" and a "lightweight ontology", as described by Guizzardi:

> On one hand, in a conceptual modeling phase in Ontology Engineering, highly-expressive languages should be used to create strongly axiomatized ontologies that approximate as well as possible to the ideal ontology of the domain. The focus on these languages is on representation adequacy, since the resulting specifications are intended to be used by humans in tasks such as communication, domain analysis and problem-solving. The resulting domain ontologies, named **reference ontologies** (...), should be used in an off-line manner to assist humans in tasks such as meaning negotiation and consensus establishment. On the other hand, once users have already agreed on a common conceptualization, versions of a reference ontology can be created. These versions have been named in the literature **lightweight ontologies**. Contrary to reference ontologies, lightweight ontologies are not focused on representation adequacy but are designed with the focus on guaranteeing desirable computational properties. (Guizzardi, 2007)

In her work, Zamborlini uses OntoUML as a language for reference ontologies and OWL as a language for lightweight ontologies. In her model-driven approach, an OWL document is built from an OntoUML domain ontology, in line with the following discussion provided by Guizzardi (we highlight some statements):

> (...) a phase is necessary to bridge the gap between the conceptual modeling of references ontologies and the coding of these ontologies in terms of specific lightweight ontology languages. Issues that should be addressed in such a phase are, for instance, determining how to deal with **the difference in expressivity** of the languages that should be used in each of these phases, or how to produced lightweight specifications that maximize specific **non-functional requirements (e.g., evolvability vs. reasoning performance)**. (Guizzardi, 2007)

Issues on expressivity are related to the logical axiomatization underlying, e.g., OntoUML and OWL specifications:

> [In OntoUML], an axiomatization in the language of intensional modal logics is incorporated in the resulting specification, constraining the interpretation of its terms. Quantified Intensional modal logics are more expressive than, for example, a SHOIN(D_n) descriptions logics, which is the language behind the formalization of OWL. (Guizzardi, 2007)

Consequently, both our work and Zamborlini's provide a transformation from OntoUML to an epistemological level language (namely, UML and OWL, respectively). Nevertheless, Zamborlini's work is not concerned with issues on information demand and (implicitly) assumes full information demand on her transformation. Her work is centered on issues about the logical axiomatization underlying an epistemological level language (e.g., expressivity, reasoning and evolvability; computational tractability and decidability). More specifically, she is mostly concerned with the representation of temporal aspects in OWL. Differently, we focus on the issues of information manipulation, i.e., addressing an information demand and identifying informational concerns and decisions. Thus, the two approaches lead to models that serve different purposes.

9.4 Efforts on Epistemological Concerns

9.4.1 Bodenreider, Smith and Burgun

Bodenreider, Smith and Burgun (Bodenreider et al., 2004) study the "intrusion of epistemology" in biomedical terminology. In the paper, they "examine the degree to which biomedical terms are created to represent not instances or classes in reality but rather features reflecting our knowledge or ignorance of such instances or classes". They provide evidence that "only some types of variant terms represent classes (universals) in reality, and that others are in fact disguised assertions about such genuine classes which are formulated as terms merely in order to meet current practical requirements of coding".

Indeed, some biomedical terms examined by the authors are not ontologically valid as universals, since their definitions are exclusively based on negation, disjunction/conjunction and/or epistemic aspects. Nevertheless, those terms could be introduced in information models as types. Thus, the discussions provided by the authors may reveal other informational concerns and decisions that were not investigated in this thesis, as we focused only on more basic concerns. In the following, we provide some insightful examples of biomedical terms that could be investigated in future works.

The authors point issues concerning "terms created in order to obtain a complete partition". Medical terminologies such as the International Classification of Diseases (ICD) aim at providing a coding system for all possible health problems. The ICD provides slots for the most frequent problems, while reserving part of the slots to grouping the less frequent diseases by means of terms involving "other". The authors examine the example of *Cystic fibrosis* in ICD-10, which has the following "subclasses":

- Cystic fibrosis with pulmonary manifestations
- Cystic fibrosis with intestinal manifestations
- Cystic fibrosis with other manifestations

- Cystic fibrosis, unspecified

According to the authors: "*Cystic fibrosis with other manifestations* is created for the purpose of representing those clinical forms not covered by the first two cases (e.g., cystic fibrosis which affects the reproductive system) and thus to complete the classification at minimal cost in extra terminological resources". In this case, terms involving "other" are not ontologically valid, since they are based on negation of properties. However, they could be introduced as types in information models in order to address scope. For instance, one may be required to know about other cases of cystic fibrosis (e.g., on reproductive system) and may aggregate data about those cases in an "other" type. Besides, the term "*Cystic fibrosis, unspecified*" is considered as a case of "vagueness, underspecification, and other hedges". Other examples include:

- *Open fracture of unspecified cervical vertebra*
- *Concussion with loss of consciousness of unspecified duration*
- *Replacement of unspecified heart valve*
- *Poisoning by unspecified drug or medical substance*
- *Colostomy not otherwise specified*

According to the authors: "Not otherwise specified expresses the – quite trivial – fact that further information could be gained but is not currently available about this particular instance" (Bodenreider et al., 2004). Terms of this sort could be used as types, addressing the ignorance of informational agents about certain aspects of reality.

A sub-case called "conjunction" is described by means of two examples. First, they present a term that is a *conjunction* of an ontologically valid term (namely, *Tuberculosis of adrenal glands*) and an epistemic aspect (namely, the method used to discover the disease). This occurs in the (rather long) term "*Tuberculosis of adrenal glands, tubercle bacilli not found (in sputum) by microscopy, but found by bacterial culture*". Second, they present terms that are *conjunctions* of two distinct (disjoint) ontological aspects (namely, skin penetration and intracranial injury). This occurs, for example, in the terms "*Closed skull fracture without intracranial injury*" and "*Closed skull fracture with intracranial injury*".

A sub-case called "modality" is exemplified by terms such as *Definite tubo-ovarian abscess*, *Probable tubo-ovarian abscess* and *Possible tubo-ovarian abscess*. According to the authors, "these qualifiers reflect the confidence of the physician at the time the diagnosis is posed, i.e., an epistemological feature that does not reflect the nature or severity of the disease being diagnosed". We conclude that such issues identified by Bodenreider et al. should be further investigated, and could influence extensions of our work, especially when concerning the treatment of vagueness, incompleteness and other forms of information imperfection.

9.4.2 ATMANSPACHER

We should note that sources from philosophy of science are relevant to the investigation of the relationship between ontology and epistemology and could also contribute to our investigation. For example, Atmanspacher has addressed this relationship in physics and philosophy of physics (Atmanspacher, 2002). As an illustration of the ontic/epistemic distinction, he cites discussions between Einstein and Bohr on quantum theory. According to Atmanspacher, "Einstein's arguments were generally ontically motivated; that is to say, he emphasized *a viewpoint independent of observers or measurements*" (Atmanspacher, 2002). By contrast, "Bohr's emphasis was generally epistemically motivated, focusing on *what we could know and infer from observed* quantum *phenomena*" (Atmanspacher, 2002). In terms of the relation between ontic and epistemic aspects, Atmanspacher says: "There are always ontic and epistemic elements to be taken into account for a proper description of a system. (...) The problem is then to use the proper level of description for a given context, and to develop and explore well-defined relations between different levels" (Atmanspacher, 2002).

The distinction in philosophy between ontic (or ontological) and epistemic (or epistemological) concerns could be target of further investigation, at least to provide insights on the distinction between the ontological level and the information level. As discussed in chapter 3, epistemological concerns are present in both levels (i.e., they are not exclusive to the information level). Nevertheless, a research on epistemology may contribute to the information level, e.g., by guiding the discovery of other informational concerns apart from the ones presented here, akin to what was done in (Bodenreider et al., 2004).

9.5 CONCLUSIONS

Evidences for the distinction between ontological and informational concerns in a two-level approach were, in a manner, envisioned by several early works. Nonetheless, those works covered the subject mostly in a theoretical manner, failing to operationalize the envisioned separation of concerns into a conceptual modeling approach.

Additionally, we provided examples demonstrating that the distinction between ontological and informational concerns is still in need of clarification, both in ontology engineering and information modeling approaches.

For future investigation, we also presented subject areas that may intersect with the discussions on this thesis (e.g., epistemology) and mentioned the contributions that information modeling may give to our characterization of the information level.

10 CONCLUSIONS

In this chapter, we present the research contributions of this thesis and discuss possible future works.

10.1 CONTRIBUTIONS

The main contributions of this thesis can be summarized as follows:

- An analysis of the role of ontology-based conceptual modeling and information modeling, providing an in-depth investigation of related works that separate or blur ontological and informational concerns;
- A two-level approach for conceptual modeling which provides a common theoretical ground to be shared by the ontological level and the information level;
- A theoretical characterization of the information level based on the nature of information and supported by ontological aspects;
- Identification and characterization of informational concerns and corresponding informational decisions, with a focus on scope, history and time tracking, reference and measurement;
- A model-driven approach for conceptual modeling, describing a transformation from OntoUML to UML, which facilitates the further design and implementation of information systems; and,
- Tool support in the Eclipse platform for the OntoUML to UML transformation, integrated in the OntoUML Infrastructure.

10.1.1 ONTOLOGY-BASED CONCEPTUAL MODELING AND INFORMATION MODELING

Throughout this thesis, we contrasted two approaches for conceptual modeling, namely, ontology-based conceptual modeling and information modeling. We have shown that the two approaches have been treated mostly in isolation and that the claimed distinction between “meaning” and “structure” is far from trivial.

In chapter 9, we presented examples in the literature of approaches that blur the distinction between modeling phenomena of interest and addressing an information demand. This confusion is especially frequent in information (or conceptual data) modeling approaches, as exemplified in the ORM language. At the same time, approaches for ontology-based conceptual modeling are not yet sufficiently clear with respect to their responsibilities. We presented an example of confusion in the field of ontology engineering, as exemplified in the DOGMA approach. Many researchers perceive the distinction between a domain ontology and an information model as merely being one of

specificity (being the former more “general” and the latter more “task specific”). As shown in this thesis, this is a rather narrow view; the actual distinction concerns the nature of information and its implications to the information level.

We also provided a detailed investigation on related works that advocate on the separation of conceptual modeling in two levels (or at least discuss some sort of separation of concerns). We concluded that, although similar ideas were developed in the literature, none of them provided a deep investigation on the subject. Thus, our investigation on the differences between conceptual modeling approaches is a valid contribution for the literature, as we helped clarifying what is actually meant by (and simply referred to as) “meaning” and “structure”.

10.1.2 Two-Level Approach

We advocated in favor of the separation of conceptual modeling in two levels, namely, the ontological level and the information level. On the one hand, we adopted the ontological level characterized by authors such as Guarino and Guizzardi. On the other hand, we characterized the information level based on several works on information (and conceptual data) modeling. We also provided our own contributions to the characterization of the information level.

In chapter 1, we provided theoretical foundations for laying a common ground to be shared by the ontological level and the information level. For that, we based our theory on the triangle of reference and on a few excerpts from Bunge’s metaphysics. The information level approach was carefully elaborated to ensure its conformance with the ontological level theory provided by Guarino and Guizzardi, so both levels could be suitably related. Ultimately, we contrasted a domain conceptualization (specified in a domain ontology) with what we called an information structure (specified in an information model). On the one hand, a domain conceptualization models a *domain* (i.e., a portion of reality) and addresses ontological concerns. On the other hand, an information structure settles the syntax of well-formed *data* about a domain and addresses informational concerns according to an information demand.

Ultimately, this thesis contributes to both the ontological level and the information level, as it clarifies the differences between these levels and establishes the responsibilities of each level. To the best of our knowledge, a similar two-level characterization is lacking in the literature. In general, we clarified the different facets of conceptual modeling in modern information systems development, as we related the relatively recent ontology-based works with the traditional information and data modeling works. In our perspective, the development process consists of the following steps: ontology-based conceptual modeling, information (or conceptual data) modeling and logical data modeling (the latter was not covered here).

10.1.3 Information Level

In order to characterize the information level, we performed a careful investigation on the definitions of “data” and “information” – a rather controversial subject – in chapters 1 and 3. We also supported our characterization with the notions of informational agent, information demand, informational concern, informational decision, information structure and information model. We took specific care to apply a uniform and clear terminology throughout this thesis; many of the investigated works did not make an effort in this respect. Besides that, the provided notions for the information level have been related to the ones of the ontological level.

In our theoretical characterization of the information level, we identified several basic informational concerns that arise from information manipulation about a certain domain. This is an important step for defining and clarifying the responsibilities that should be attributed to the information level, as we studied in detail the nature of information manipulation. In depth, we revisited the triangle of reference in chapter 3 and identified two sorts of informational concerns, namely, information demand concerns and representation concerns. Information demand concerns take into account what the informational agent is required to know about phenomena of interest. The set of concerns we have identified includes scope, history tracking, time tracking and measurement. In contrast to information demand concerns, representation concerns determine how information should be encoded in data. Among those concerns are the selection of the information modeling technique, the lexicalization of reference schemes and data types.

Although those concerns are frequently addressed in the information modeling literature, works do not usually provide an in-depth theoretical investigation on the reasons why these concerns arise and why information modeling should address them. That is to say, the responsibilities of the information level are not always made clear. At the same time, some authors provide theoretical studies on information, but do not explicitly apply them in an information modeling technique. In this thesis, we have dealt with both aspects, i.e., we investigated the nature of information and applied the theoretical perspective in information modeling.

10.1.4 Informational Decisions

A significant contribution of this thesis is the identification of informational decisions for each informational concern, in special, the elaboration of those decisions in a systematic correspondence to the ontological level theory. Ultimately, informational decisions are the means by which the ontological level and the information level are systematically related. They support the parameterization of our model-driven approach, in which different decisions (parameters) produce different information models. Information (and conceptual data) modeling approaches usually present similar decisions, but in a non-systematic manner, which means informational decisions are

treated as any other modeling decisions and are all addressed in a single level, mixing ontological and informational concerns.

We have shown that informational concerns are *not trivially* related to meta-categories, i.e., different concerns are related to different meta-categories. We present the summary of informational concerns, decisions and related ontological aspects in Table 5.1. In our model-driven approach, the structure of an information model takes into account the structure of a domain ontology, as discussed in chapters 4 and 5, involving all meta-categories at the ontological level (viz. Kind, SubKind, Category, Relator, Role, Role Mixin and Quality). As discussed in chapter 5, the "side effects" of scope decisions are governed by the meta-categories at the ontological level and make use of the ontological notions of rigidity and principle of identity. As discussed in chapter 6, history and time tracking are related to the same meta-categories (viz. Kind, Relator and Quality). As discussed in chapter 7, reference is related to several meta-categories (viz. Kind, Relator and Category), while measurement is exclusively related to one meta-category (viz. Quality).

Consequently, the rich ontological distinctions at the ontological level assisted our investigation of informational concerns at the information level. In particular, the ontological theories were considerably important for characterizing history and time tracking in chapter 6 and for contrasting reference and measurement in chapter 7. This provides some evidence for the usefulness of the ontological level theories, as it presents an example on how ontological distinctions can support theoretical investigation.

Informational Concern	**Informational Decision**	**Related Ontological Meta-Categories**
Scope	Information on instances of certain universals	All meta-categories
History Tracking	Information on present, past or both	Kind, Relator, Quality
Time Tracking	Information on start time, end time and duration	
Reference	Lexicalization of reference schemes	Kind, Relator, Category
Measurement	Lexicalization of quality structures	Quality

Table 10.1 - Summary of informational concerns and decisions, and the corresponding ontological aspects

Moreover, our research revealed that addressing informational concerns is a considerably complex task and that each concern may be addressed differently according to information demands. Thus, we provided evidence that informational concerns should not be handled in a single level along with ontological concerns. By establishing relations between the ontological and the information levels, we have aimed at leveraging the benefits of ontological level distinctions, while addressing the unavoidable informational decisions. The advantage of the separation into levels is that an unbiased

domain ontology is likely to be shared by a larger community, which may agree on ontological concerns but may have different information demands. This can be the basis for the establishment of semantic interoperability; in this case, the domain ontology should be considered as a reference model for interoperation.

10.1.5 Model-driven Approach and Information Modeling Technique

We applied our theoretical study on informational decisions and ontological aspects in a model-driven approach for conceptual modeling, in which a domain ontology is used as a starting point for the specification of an information model. More specifically, we elaborated a model transformation from OntoUML to UML, in an object-oriented approach for information modeling. By means of several examples, we illustrated how informational decisions impact the structure of information models. For a given domain ontology, we depicted the multitude of conformant information models that address different information demands with respect to the considered domain.

In our approach, we acknowledged the role of an information model as a specification to be further used in the design and implementation of information systems. Consequently, we advanced some commitments that are commonly addressed during those phases. By doing such, we contributed to bridging the gap between a domain ontology and a logical data model. More specifically, we opted for an object-oriented approach for information modeling that avoids dynamic classification. This required considerable effort, as domain ontologies written in OntoUML rely on dynamic classification of entities in reality.

As a result, we presented in chapter 4 an investigation into the representation of role playing in object-oriented modeling. We have shown that, although the subject has been considerably investigated in the literature, none of the approaches suited our purposes. The previous approaches did not usually acknowledge on their representations that a Role (e.g., Student) is dependent on other universals (e.g., Enrollment and University). As a consequence, they usually restricted themselves to the representation of data structures on partial information demand, that is to say, when one is only partially interested in role playing aspects (e.g., only data about the Student, but not on the University). Thus, we provided our own approach for the representation of data on roles in information modeling; one that leverages the OntoUML representation of Roles as relationally dependent universals mediated by Relators. Our approach is able to cover both situations of partial and full information demand with the same information modeling technique, namely, representing relator types instead of role types.

In line with the discussions of chapter 4, we presented informational decisions concerning scope in chapter 5. We considered that every universal in the domain ontology may either be inside or outside the scope of the information demand. We have provided several examples which show

the impacts of scope decisions on information models. Basically, when a universal in a domain ontology is outside the scope, some corresponding constructs in the information model will be absent (types, associations, generalizations, attributes). In addition, we considered that a scope decision on some universals may impact scope decisions on other universals, as a "side effect". For example, we assumed that the scope of a Relator impacts the scope of the mediated Roles. We have taken this approach, as the other alternatives had several complex impacts in the information model, such as the migration and duplication of attributes and associations.

In chapter 6, we described our model-driven approach for history and time tracking. We addressed history tracking by means of the "current" attribute, in order to avoid dynamic classification at the information level. In addition, we considered that history tracking decisions affect the cardinalities of associations involving relator types. We addressed time tracking via "start", "end" and "duration" attributes. This involved the introduction of optional cardinalities for the "end" attribute. Moreover, we assumed that time tracking decisions also incorporate measurement decisions, since data type specifications for time instants and time durations are required.

In chapter 7, we presented the model-driven patterns for reference and measurement. Although we argued that those concerns are fundamentally distinct, both were addressed via attributes and data types. We also discussed scope and history tracking decisions for Qualities. When history tracking is performed, a measure type is created in the information model. Moreover, we considered the usage of optional cardinalities for measurement attributes.

We summarize the impacts produced by decisions in information models in Table 10.2.

Informational Concern	**Impact on the information model**
Information Modeling Technique / Scope	types (classes), associations, generalizations, attributes, optional cardinalities
Reference	reference attributes (identifiers) and data types
Measurement	measurement attributes and data types, optional cardinalities
History Tracking	"current" attribute, relator type cardinalities, measure type
Time Tracking	"start", "end", "duration" attributes, optional cardinalities

Table 10.2 - Summary of the impact of informational decisions on information models

10.1.6 TOOL SUPPORT

As described in chapter 8, considerable effort was made to provide a tool implementation for the model-driven approach presented in this thesis. We have demonstrated the feasibility of our model-driven approach by developing the Onto2Info plug-in. We have shown that informational decisions can be systematically captured by means of a graphical user interface consisting of several tabs for each informational concern.

We implemented the model transformation as a component of the OntoUML Modeling Infrastructure (Carraretto, 2010), which is developed in the Eclipse environment using the Eclipse Modeling Framework (EMF) and the Ecore metamodeling language. The OntoUML Infrastructure is well-established and incorporates the OntoUML reference metamodel, which is currently the most complete metamodel implementation of the OntoUML language. Moreover, the infrastructure incorporates several other model transformations that are related to OntoUML such as OntoUML to Alloy (Benevides, 2010) and OntoUML to OWL (Zamborlini, 2011). Hence, by providing a transformation from OntoUML to UML, our work adds value to the OntoUML Infrastructure.

10.2 FUTURE WORK

We have not aimed at an exhaustive set of informational decisions and concerns, and many relevant informational decisions could be reported in future works. These include those decisions that arise from inferred information (which may be obtained deductively and inductively with different consequences), from the limited and incomplete knowledge about phenomena of interest.

As mentioned in chapter 9, we refrained from addressing derived constructs in information models (types, relations and attributes). The definition of information model constructs (data structures) may involve innumerable kinds of information manipulation (e.g., conjunction, disjunction, negation, averages, probability, counts), making it difficult to enumerate all the possible informational decisions. At the same time, derived data structures have considerable importance for convenient information manipulation over a domain and they play an important role in addressing an information demand. Therefore, derived structures should be investigated in the future.

For future work, we could investigate the distinction between ontological and epistemological concerns, with the intent to gain insights on informational concerns. In chapter 9, we cited some examples provided by (Bodenreider et al., 2004) and highlighted the ones concerning the treatment of vagueness, incompleteness and other forms of information imperfection.

We have not explored in depth the so-called meta-data, frequently addressed in the data quality literature (Wand & Wang, 1996). We restricted ourselves to mentioning how meta-data could contribute to addressing an information demand on measurement events and baptism ceremonies in

chapter 7. This could also be a topic for further investigation. One kind of meta-data that may be of particular interest is “data provenance” which “pertains to the derivation history of a data product starting from its original sources” (Simmhan, Plale, & Gannon, 2005). For example, data provenance is important for investigating the source of data generated by complex manipulations (e.g., workflows) and of data stored in data warehouses or in datasets available to the public domain.

The way we addressed the informational concerns identified here could be considerably improved based on the sophisticated support offered by works on information and conceptual data modeling. For instance, we mentioned the ORM support for complex reference schemes, which could improve our addressing of reference and measurement.

As we identified two sorts of informational concerns (viz. information demand and representation concerns), we could explore a division of our approach in two steps. The first step would initially address information demand concerns in a manner independent of data. Then, the second step would address representation concerns (including the information modeling technique) and, consequently, would commit to certain design decisions. In this thesis, we have not considered alternatives for design decisions in the resulting information models, as we focused on the addressing of informational decisions. In future work, we could explore design decisions and introduce them as parameters of the model-driven approach, as we have done with informational decisions. Besides, our model-driven approach could be adapted to create transformations from a domain ontology to semantic web artifacts. This means we could adapt our approach and our tool to generate OWL documents and RDF schemas addressing an information demand, instead of UML class diagrams.

The separation into levels could be used to provide a methodology that guides the development of a domain ontology that represents the underlying semantics of an information model, in a bottom-up approach. In fact, a bottom-up approach could be used to investigate other informational concerns and decisions, by exploring common patterns in information models and attempting to find their ontological interpretation. As an illustration, a frequent example presented in the information modeling literature is that of representing gender as an attribute whose instances are either “M” (standing for “male”) or “F” (standing for “female”). At the ontological level, we represented gender as the Man and the Woman SubKinds of the Person Kind. Thus, this could be seen as a lexicalization of universals at the ontological level as attributes and enumerations at the information level. This methodology could be used to reveal other informational concerns as well.

Besides that, our usage of the ontological level was not exhaustive. In this thesis, we only tackled the most important concepts of UFO according to our perspective. For future work, one could extend our approach to the remaining portions of the UFO (quantities, collectives, phases, semi-rigid mixins, modes, formal relations, part-whole relations, etc.).

Finally, the approach and the tool should be subject to application in large-scale projects which could provide some feedback concerning usability and applicability of the results of this work.

REFERENCES

Ashenhurst, R. (1996). Ontological Aspects of Information Modeling. *Minds and Machines*, *6*(3), 287–394.

Atmanspacher, H. (2002). Determinism is Ontic, Determinability is Epistemic. In H. Atmanspacher & R. Bishop (Eds.), *Between Chance and Choice: Interdisciplinary Perspectives on Determinism* (pp. 49–74). Thorverton, England: Imprint Academic.

Benevides, A. B. (2010). *A model-based graphical editor for supporting the creation, verification and validation of OntoUML conceptual models*. Federal University of Espírito Santo, Vitória.

Bodenreider, O., Smith, B., & Burgun, A. (2004). The Ontology-Epistemology Divide: A Case Study in Medical Terminology. In A. C. Varzi & L. Vieu (Eds.), *3rd International Conference on Formal Ontology in Information Systems (FOIS 2004)* (pp. 185–195). Turin: IOS Press.

Bunge, M. (1977). *Ontology I: The Furniture of the World*. D. Reidel Publishing Company.

Cabot, J., & Raventós, R. (2004). Roles as Entity Types: A Conceptual Modelling Pattern. *23rd International Conference on Conceptual Modeling (ER'04)* (pp. 69–82).

Carraretto, R. (2010). *A Modeling Infrastructure for OntoUML*. Federal University of Espírito Santo, Vitória.

Carraretto, R., & Almeida, J. P. A. (2012). Separating Ontological and Information Modeling Concerns: Towards a Two-Level Model-Driven Approach. *16th IEEE International Enterprise Distributed Object Computing Conference Workshops* (pp. 29–37). Los Alamitos, CA: IEEE Computer Society Press.

Chen, P. P.-S. (1976). The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems*, *1*(1), 9–36.

Floridi, L. (2010). *Information: A very short introduction*. Oxford University Press.

Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. (G. Booch, I. Jacobson, & J. Rumbaugh, Eds.) (3rd ed.). Boston: Addison-Wesley.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., & Schneider, L. (2002). Sweetening ontologies with DOLCE. *Lecture Notes in Computer Science*, *2473*, 166–181.

Gottlob, G., Schrefl, M., & Röck, B. (1996). Extending object-oriented systems with roles. *ACM Transactions on Information Systems*, *14*(3), 268–296.

Grenon, P., Smith, B., & Goldberg, L. (2004). Biodynamic ontology: applying BFO in the biomedical domain. *Studies in health technology and informatics*, *102*(ii), 20–38.

Gruber, T. R. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies*, *43*(5-6), 907–928.

Guarino, N. (1994). The Ontological Level. In R. Casati, B. Smith, & G. White (Eds.), *Philosophy and the Cognitive Sciences* (pp. 443–456). Vienna: Hölder-Pichler-Tempsky.

Guarino, N., & Welty, C. (2002). Evaluating ontological decisions with OntoClean. *Communications of the ACM*, *45*(2), 61–65.

Guizzardi, G. (2005). *Ontological foundations for structural conceptual models*. University of Twente, Enschede.

Guizzardi, G. (2007). On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. In O. Vasilecas, J. Eder, & A. Caplinskas (Eds.), *Databases and Information Systems IV - Selected Papers from the Seventh International Baltic Conference DB&IS'2006* (pp. 18–39). Amsterdam: IOS Press.

Guizzardi, G., Masolo, C., & Borgo, S. (2006). In Defense of a Trope-Based Ontology for Conceptual Modeling: An example with the foundations of Attributes, Weak Entities and Datatypes. In D. W. Embley, A. Olivé, & S. Ram (Eds.), *25th International Conference on Conceptual Modeling (ER'06)* (pp. 112–125). Tucson, AZ, USA.

Gärdenfors, P. (2000). *Conceptual Spaces: The Geometry of Thought* (1st ed.). The MIT Press.

Halpin, T., & Morgan, T. (2008). *Information modeling and relational databases* (2nd ed.). Morgan Kaufmann.

Heller, B., & Herre, H. (2004). Ontological Categories in GOL. *Axiomathes*, *14*(1), 57–76.

Jarrar, M., & Meersman, R. (2009). Ontology Engineering - The DOGMA Approach. In E. Chang & K. Sycara (Eds.), *In Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web* (1st ed.). Springer.

Kent, W. (2000). *Data and Reality* (2nd ed.). 1st Books Library.

Kripke, S. A. (1980). *Naming and necessity*. Cambrige, Massachusetts: Harvard University Press.

Langefors, B. (1980). Infological models and information user views. *Information Systems*, *5*(1), 17–32.

Mill, J. S. (1882). *A System of Logic, Ratiocinative and Inductive* (8th ed.). New York: Harper & Brothers.

Moore, G. E. (1953). *Some main problems of philosophy* (Vol. 1). Routledge.

Mylopoulos, J. (1992). Conceptual Modelling and Telos. *Information Systems Journal*, 1–20.

Mylopoulos, J. (1998). Information Modeling in the Time of the Revolution. *Information systems*, *23*(3-4), 127–155.

Newell, A. (1982). The Knowledge Level. *Artificial Intelligence*, *18*(1), 87–127.

OED. (2009). Oxford English Dictionary Second Edition on CD-ROM (v. 4.0).

OMG. (2006). Object Constraint Language (OCL), OMG Available Specification, version 2.0.

OMG. (2008). Semantics of Business Vocabulary and Business Rules (SBVR), OMG Available Specification, version 1.0.

OMG. (2011). OMG Unified Modeling Language (OMG UML), Superstructure, version 2.4.

Ogden, C. K., & Richards, I. A. (1923). *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Routledge & Kegan Paul.

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The Unified Modeling Language Reference Manual*. Reading, Massachusetts: Addison Wesley Longman, Inc.

Simmhan, Y. L., Plale, B., & Gannon, D. (2005). *A Survey of Data Provenance Techniques. Technical Report IUB-CS-TR618.* Computer Science Department, Indiana University.

Simsion, G. C., & Witt, G. C. (2005). *Data Modeling Essentials*. San Francisco, CA: Morgan Kaufmann Publishers.

Steimann, F. (2000). On the representation of roles in object-oriented and conceptual modelling. *Data & Knowledge Engineering*, *35*(1), 83–106.

Steinberg, D., Budinsky, F., Paternostro, M., & Merks, E. (2008). *EMF Eclipse Modeling Framework*. (E. Gamma, L. Nackman, & J. Wiegand, Eds.) (2nd ed.). Boston: Addison-Wesley Professional.

Strawson, P. F. (1950). On Referring. *Mind*, *59*(235), 320–344.

Wand, Y., & Wang, R. Y. (1996). Anchoring Data Quality Dimensions in Ontological Foundations. *Communications of the ACM*, *39*(11), 86–95.

Wieringa, R., de Jonge, W., & Spruit, P. (1995). Using dynamic classes and role classes to model object migration. *Theory and Practice of Object Systems*, *1*(1), 61–83.

Zamborlini, V. C. (2011). *Estudo de Alternativas de Mapeamento de Ontologias da Linguagem OntoUML para OWL: Abordagens para Representação de Informação Temporal*. Federal University of Espírito Santo, Vitória.