The Pennsylvania State University

The Graduate School

College of Information Sciences and Technology

TOWARDS ONTOLOGY-DRIVEN INFORMATION SYSTEMS: GUIDELINES TO THE CREATION OF NEW METHODOLOGIES TO BUILD ONTOLOGIES

A Dissertation in

Information Sciences and Technology

by

Andrey Soares

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2009

The dissertation of Andrey Soares was reviewed and approved* by the following:

Frederico Fonseca
Associate Professor of Information Sciences and Technology
Dissertation Adviser
Chair of Committee

Mary Beth Rosson
Professor of Information Sciences and Technology

David Hall
Professor of Information Sciences and Technology

Timothy Simpson
Professor of Mechanical Engineering and Industrial Engineering

John Yen
Professor of Information Sciences and Technology
Associate Dean for Research and Graduate Programs

*Signatures are on file in the Graduate School.

ABSTRACT

This research targeted the area of Ontology-Driven Information Systems, where ontology plays a central role both at development time and at run time of Information Systems (IS). In particular, the research focused on the process of building domain ontologies for IS modeling.

The motivation behind the research was the fact that researchers have not yet produced comprehensive guidelines for building ontologies for IS. A recent survey reported that 60% of the respondents did not use any methodology to build their ontologies. Ontology engineering is still considered an art, rather than and engineering activity. The results of our preliminary research on building an ontology for a given domain revealed four important issues related to Ontology-Driven Information Systems. These issues are related to metamodels, procedural knowledge, temporal relations and knowledge acquisition.

Based on these concerns, we set up a research to investigate existing methodologies that could provide principled guidelines to build ontologies and to overcome the issues raised in the preliminary study. We searched major bibliographic databases from which we selected 30 methodologies to investigate. The analysis of the methodologies was formulated around the core components of an ontology and the four issues raised in our preliminary research. We also discussed the methodological features that are relevant to the process of building ontologies for Information Systems.

Our final results confirmed the four issues among the methodologies analyzed. Besides, axiomatization has emerged as another important issue for Ontology-Driven Information Systems. Moreover, the frequent use of scenarios in the initial steps of the methodologies motivated us to further investigate their use in building ontologies. We proposed to use the components of a scenario as the ontological constructs of a metamodel ontology. To illustrate the use of scenarios in the building of domain ontologies, we developed a proof-of-concept experiment. The experiment successfully showed that a scenario-based approach can help acquiring and representing relevant domain knowledge to be used in IS modeling, and can be used to improve the methodologies used to build ontologies.

TABLE OF CONTENTS

List of Figures

List of Tables

Acknowledgements

I would like to express my deepest gratitude to everyone who has directly or indirectly contributed to this dissertation, especially to:

My advisor, mentor and friend, Dr. Frederico Fonseca, without whom this dissertation would not be possible. I have learned a great deal from him about my professional and personal development. I am grateful for his confidence, guidance, encouragement, and for keeping me motivated and on track. I also thank him for introducing me to the area of Ontology. You have been a great inspiration.

My dissertation committee members, Dr. Mary Beth Rosson, Dr. David Hall and Dr. Timothy Simpson. Their insightful comments and questions helped me to shape and refine this work. I also would like to thank Dr. Steve Sawyer, a former member of my dissertation committee, for the discussions and feedback about my research.

My wife Cris, for her constant support throughout the years even before I started this journey. She has been a continuing source of love and inspiration. Thank you very much for being always by my side, for believing in me, for your patience, and for making this dissertation possible. I could not have done it without you.

My parents Luiz and Neuza, who taught me the value of education and inspired me to purse my academic goals.

All of the professors, staff, students, and friends from the IST. A special thanks to David Saab, Fabiano Beppler, Jan Mahar, Lisa Lenze, Betty Blair, Rhonda Boonie and Sue Van Vactor for their friendship and support.

Chapter 1.
INTRODUCTION

The increasing use of ontologies in Information Systems (IS) and the proliferation of conceptual modeling methods lacking theoretical foundations (Rosemann & Wyssusek, 2005) brought the attention of researchers to investigate theories of ontology in Information Systems. Despite three decades of research and a shared understanding that ontology plays a central role in Information Systems (Bubenko, 1979; Fonseca, 2007; Guarino, 1998; Wand & Weber, 1989), researchers have not yet produced comprehensive guidelines for building ontologies for Information Systems Analysis and Design (ISAD) (Guarino & Welty, 2000; Yildiz & Miksch, 2007).

Guarino (1998) proposed, in his seminal paper, the concept of Ontology-Driven Information Systems (ODIS) where he envisioned the use of ontologies in two distinct stages of Information Systems: (1) at development time, and (2) at run time. At development time, an ontology can be used in the conceptual modeling phase of IS, representing the knowledge of a given domain and supporting the creation of IS components. At run time, an ontology can be used as another part of the information system driving all of its aspects and components, that is, the system runs in accordance to the content of the ontology (Uschold, 2008). The most explored use of ontologies in IS is at run time, nevertheless several authors have focused their research on the theories and the use of ontologies in IS at development time (Fonseca, 2007; Green & Rosemann, 2005; Guarino, 1998; Guizzardi, 2005; Kishore *et al.*, 2004; Wand & Weber, 1990; Wyssusek, 2004). Ontology-Driven Information Systems is discussed in more detail in Chapter 2.

This research focuses on Ontology-Driven Information Systems at development time (Soares & Fonseca, 2007), which can be seen as a two-phase framework (Figure 1). Phase 1 focuses on the construction of ontology as an artifact to represent the knowledge of a given universe of discourse for its use in IS modeling. Phase 2 focuses on the use of ontology to support the creation of IS components, namely application programs, information resources, and user interfaces (Guarino, 1998). The ontology in this framework is the backbone of the conceptual modeling activities supporting the modeling of “two different domains: reality and the information system” (Wand & Weber, 1989, p.82). According to Mylopoulos (1992), conceptual modeling refers to “the activity of formally describing some aspects of the physical and social world around us for purpose of understanding and communication” (p.3).

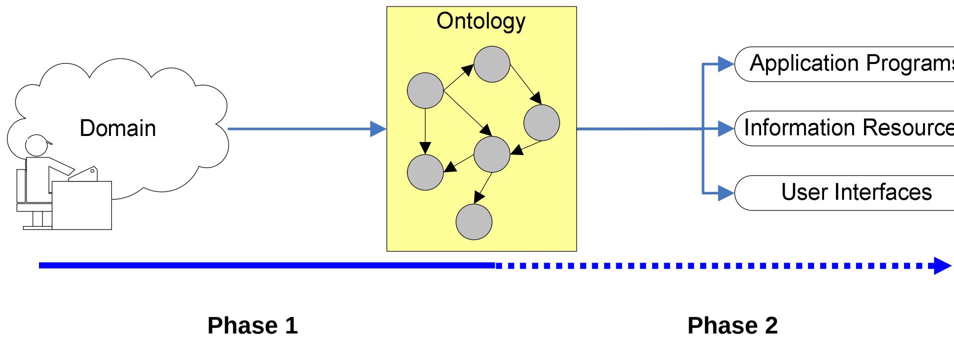

Figure 1: Ontology-Driven Information Systems at development time

An important reason for employing ontologies at development time is that “when domain and task ontologies are used during development time, the semantic content about the domain contained within those ontologies can be easily transformed and translated into IS components, thereby enabling knowledge reuse, reducing cost of conceptual analysis, and assuring the ontological adequacy of the IS” (Kishore *et al.*, 2004). A closer look at the ontological constructs and the relationship between them can uncover the IS components of the system under investigation. After all, ontology represents the knowledge of a given domain, which should be analogous to the knowledge used by designers in their activities (Fonseca, 2007; Zlot *et al.*, 2002).

The initial idea for this research was to investigate how ontologies drive ISAD activities and help the creation of IS components. However, doing this would be assuming that ontologies are properly created. Nevertheless, if ontologies are going to be used to help the creation of IS components, the research should first inquire about which ontologies are appropriate to be used in ISAD, or how to create ontologies that represent the knowledge needed for modeling an IS. Thus, the research focuses on Phase 1 (Figure 1) to investigate existing methodologies to build ontologies that are suitable to Information Systems.

1.1 Motivation

The first step in the conceptual modeling activities of Information Systems is the transformation of the perceived real-world into a model of the world it intends to represent (Wand & Weber, 1989). According to Wand & Weber (2004, p.xii), because ontology is used to represent the real-world, “descriptions [of the world] will only be as good as our ontologies”, and because Information Systems are models of real-world systems, they also “will only be as good as our ontologies”. This position advocates that building better ontologies should improve the process of designing Information Systems and consequently the quality of the system delivered. However, building ontologies for Information Systems is not an easy task, and requires a great set of skills from the ontology engineer.

Ontologies have been used across different domains and for different purposes (Guarino, 1998; McGuinness, 2001), nonetheless, no methodological approach for building ontology has been prominent. A recent survey (Cardoso, 2007) shows that 60% of the participants did not use a methodology to develop ontologies (Figure 2).

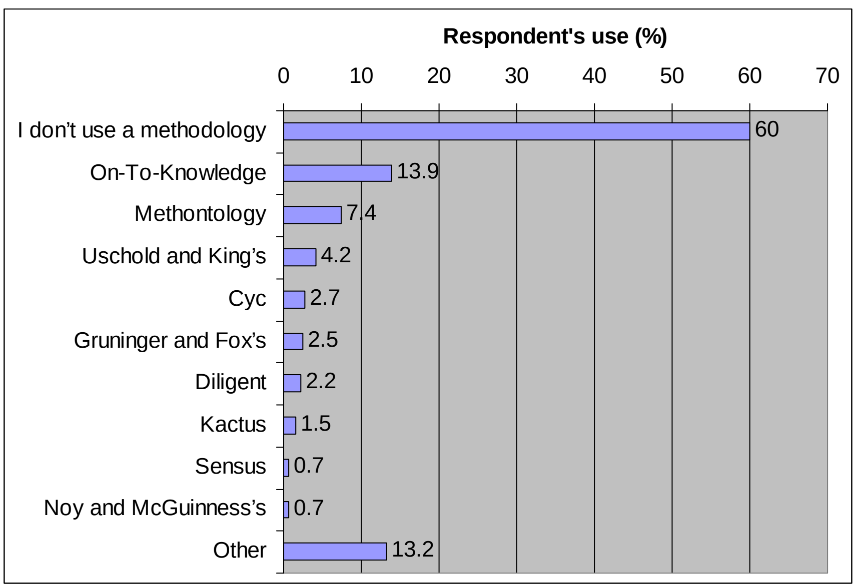

Figure 2: Methodologies used to develop ontologies (from Cardoso, 2007, p.87)

Gómez-Pérez et al., referring to the mid-1990s, explain that “the absence of common and structured guidelines slowed the development of ontologies within and between teams, the extension of any ontology, the possibility of ontologies of being reused in others, and the use of ontologies in final applications" (Gómez-Pérez *et al.*, 2004, p.107). Approximately a decade later, Cardoso's survey shows that the lack of methodological approaches still remains; and, the process of building ontologies is still considered an art rather than an engineering process (Abou-Zeid, 2003; Gómez-Pérez *et al.*, 2004; Jin *et al.*, 2004; Peralta *et al.*, 2005b; Sugumaran & Storey, 2002; Uschold, 1996; van der Vet & Mars, 1998).

This research is a contribution to shorten the gap in ontology design by investigating methodologies to build ontologies from scratch, and seeking methodological principles that can guide the design of ontologies suitable for IS modeling.

1.2 Scope and problem space

This research extends Rolland and Prakash's (2000) two-phase organization of system life-cycle (i.e., conceptual modeling and systems engineering) to illustrate the view of ontology as an important IS artifact responsible for representing the knowledge of a given domain, and for providing knowledge during the creation of conceptual schemas (Figure 3). In this case, ontology is neither a substitute nor a competitor of existing conceptual schemas, but a significant and complementary resource that can provide formalized knowledge about the universe of discourse (Fonseca & Martin, 2005; Wand *et al.*, 1995).

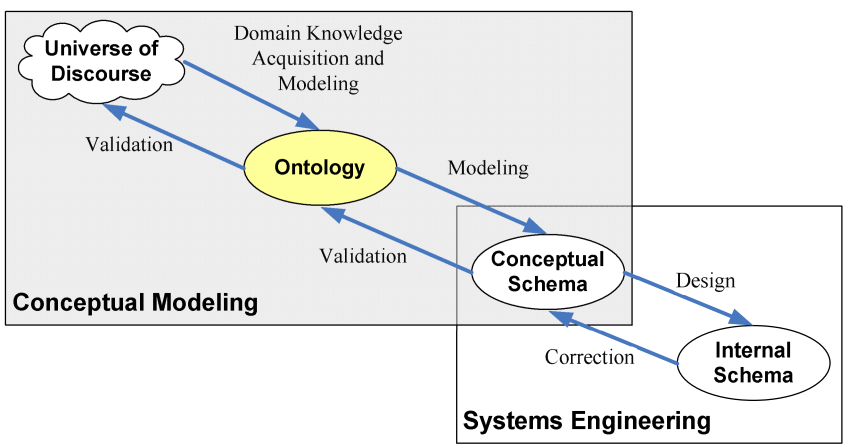

Figure 3: The role of an ontology in Information Systems (adapted from Rolland & Prakash, 2000)

As a matter of fact, ontology is not a new thing in the process of designing Information Systems. System designers often rely on ontological questions to build a conceptual model (Guarino & Guizzardi, 2006), even though they are not always aware of the ontological nature of these questions. Examples of ontological questions are (Guizzardi, 2005): Is there a unique identifier for all objects? Is this a whole-part relationship? Is this a property of an object? Thus, ontology is not often explicitly used as an IS artifact in the development process of IS. Instead, what happens is the creation of conceptual schemas that are based on the knowledge produced, and not yet formalized, during the conceptual modeling phase of IS.

Ontologies are different from conceptual schemas (Fonseca & Martin, 2007). Ontology focuses on the universe of discourse and is used to guide the creation of conceptual schemas, while a conceptual schema focuses on (and is limited by) the information system being modeled (Evermann, 2004; Fonseca & Martin, 2007). Moreover, conceptual schemas carry and represent a fraction of the knowledge about the domain. Therefore, in order to create a conceptual schema, only the knowledge that is suitable for that specific schema will be used. An Entity-Relationship schema, for example, only needs three constructs to be represented: Entities, Attributes, and Relations (Dennis *et al.*, 2005). If not used for representing another conceptual schema, or formalized in another way, the remaining knowledge remains tacit in the minds of users and designers; and, the knowledge already acquired would be neither transferable nor reusable in other IS design phases.

This research focuses on the conceptual modeling phase of an information system life-cycle, shown in Figure 3. Within that scope, the research is particularly interested in the process of domain knowledge acquisition and modeling, which is building ontologies suitable for IS modeling. The research shares the view that every information system embeds knowledge about some application and its respective domain (Sowa, 2000), which means it has its own ontology (Guarino, 1998). However, it argues that during the conceptual modeling of a system, the ontology embedded in the designers and domain experts' mind many times is not properly represented as an IS artifact.

1.3 Research objective and questions

The objective in this research is to identify methodological guidelines for the process of "building ontologies from scratch" (Pinto & Martins, 2004, p.41) that are suitable to IS modeling. In particular, we seek, among existing methodologies, specific approaches that can be employed to develop ontologies for ISAD, and we propose a Scenario-based approach to improve the process of building such ontologies.

These goals are addressed by the following research question (RQ):

- *RQ: How can we build ontologies that are appropriate to IS modeling?* This high level question refers to the central role of ontology in Information Systems as described in the two-phase framework (see Figure 1). This question focuses on Phase 1, which refers to the process of creating such ontologies.

Based on the issues (i.e., knowledge acquisition, metamodels, procedural knowledge, and temporal relations) identified in the preliminary research (see Chapter 3), the high-level question is partitioned into more specific sub-questions to address the issues with regard to the building of ontologies suitable to Information Systems:

- *RQ_1: How can we acquire knowledge about a domain?* The main purpose of ontologies is to represent knowledge about a given domain. This question aims to identify approaches to guide experts and designers in the process of acquiring knowledge relevant to IS.
- *RQ_2: How can we identify and represent procedural knowledge?* This question investigates the approaches proposed by existing methodologies for describing the behavior of a system, which is usually represented by events and tasks. The procedural knowledge refers to a sequence of tasks needed to achieve a goal, that is, the description of how to do things (Milton, 2007).
- *RQ_3: How can we identify and represent temporal relations?* Tasks and events are related to each other through time intervals (Allen, 1983), such as before and after. This question identifies the methodological approaches used to identify and represent domain knowledge related to time constraints.

- *RQ_4: How can we use meta-ontology to guide the creation of domain ontologies for IS modeling?* By agreeing on a meta-ontology (i.e., a high-level model of the domain), designers would establish their ontological commitment to a particular view of the domain under investigation. The constructs of a metamodel ontology work as a frame for domain ontologies as well as a guideline to build them (Gómez-Pérez, 1998). This question attempts to identify underling structures (i.e., metamodel) to be used for representing knowledge needed in the process of IS modeling.

1.4 Dissertation organization

The remainder of this dissertation is organized as follows.

Chapter 2 presents the underlying background that supports the research on the use of ontology in Information Systems, especially in the Ontology-Driven Information Systems area. The chapter discusses conceptual modeling, ontology, knowledge representation, and scenarios.

Chapter 3 discusses the results of our preliminary investigation building ontologies for a given domain. The chapter describes the four issues (i.e., metamodel, procedural knowledge, temporal relations, and knowledge acquisition) related to the development of domain ontologies for IS modeling.

Chapter 4 describes the research methodology used in this dissertation. The methodology is based on the method of Systematic Reviews. The guidelines used in the review are described, and the procedures to identify, to select and to analyze the primary studies (i.e., papers) are detailed. The chapter also includes a list of 30 methodologies to build ontologies that have been selected for review, and a list of twelve categories that we developed to analyze the methodologies.

Chapter 5 synthesizes the methodologies selected for review. The methodologies are discussed with regard to the categories developed in Chapter 4. The main features of the methodologies are described to illustrate the categories.

Chapter 6 discusses the lessons learned from the analysis of the methodologies, and the findings related to the issues identified in Chapter 3. A description of the main methodological approaches addressing the issues is presented. An approach based on scenarios is proposed to improve the process of building ontologies for Information Systems.

Chapter 7 presents conclusions and future work. A summary of the dissertation and its major findings are described. Its contributions to practitioners and researchers are discussed; and, the future research directions are presented.

Appendix A presents a summary of the analysis of the methodologies, including the following criteria: Knowledge Acquisition, Identify Concepts/Relationships, Identify Tasks, Identify Temporal Relations, Identify Axioms, Ontology Levels, and Mapping between levels.

Chapter 2.
BACKGROUND

The use of ontology in the conceptual modeling process of IS, as suggested in this research, is not a new idea. Solvberg (1979) proposed that "the conceptual schema should contain an ontological subschema (i.e., a 'reality' model)" (p.111). Bubenko (1979) also suggests that a conceptual schema should be preceded by an understanding model of the reality, which is in someone's mind even before they formulate the requirements of a system. He calls it a conceptual information model, and he adds that this "is not the same as what eventually will be stored and maintained by a data based management system" (p.130). From his observations, Bubenko concludes that creating a conceptual schema demands not only the systems requirements but also implicit knowledge about the reality, which are not part of the schemas. Nevertheless, he argues that "the value of a conceptual information model lies rather in its ability to give guidance to design a correct and consistent conceptual data base and processing model" (p.137). The view of ontology having a central role in the conceptual modeling phase of an IS, promoted in this research, is in line with Bubenko's conceptual information model.

At that time, Bubenko justified why the understanding model was being ignored in theoretical and practical work. First, because "it was probably easier and more natural, for a computing-trained designer, to specify an algorithmic solution to a problem than to formally document his thoughts on requirements, assumptions, relations and conditions which constitute the 'mental basis' for the algorithmic solutions" (p.138). Second, because the understanding model was still under development and there were not many notational approaches to represent the model. Three decades later, Bubenko's concerns still remain. However, the field of Ontology Engineering is more mature and should provide several alternatives with respect to representing the understanding model (i.e., the ontology).

The use of ontology advocated in this research is twofold: first, a mechanism to represent the domain knowledge for IS modeling, and second, a frame (i.e., metamodel) that will provide a guide to capture and to represent the domain knowledge. Among a variety of theoretical and practical approaches in the definition of principles and methods for ontology development, the following work have directly influenced this research. First, Guarino (1998) coined the term Ontology-Driven Information Systems, which is the main framework driving this research. Second, Wand & Weber (1989) explored theories of ontology, in terms of a grammar to describe the real-world for the purpose of IS modeling. The extension, critique, and discussion of the BWW (Bunge-Wand-Weber) ontology contributed to advance the field (Lyytinen, 2006) and to drive the interest of researchers to ontology at the conceptual modeling of IS. Upon trying to answer the question "*How can we model the world to better facilitate our developing, implementing, using, and maintaining more valuable information systems?*" (Wand & Weber, 2002, p.373), Wand & Weber identified important research opportunities related to conceptual modeling. Finally, Guizzardi (2005) compiled a comprehensive material regarding the ontological foundations for structural conceptual models. This research builds upon their work and adds knowledge about methodological approaches to build ontologies that are suitable for IS modeling.

Next we review the underlying concepts used in this research. We start discussing the conceptual modeling phase of an Information Systems life-cycle (Figure 3), and in particular, the use of ontology as an IS artifact used to model a domain. We also discuss characteristics of knowledge engineering related to acquisition and representation of domain knowledge; and, we finish the chapter talking about scenarios, which we propose as a key feature to be used in the building of ontologies for IS.

2.1 Conceptual modeling

Much research has been conducted on conceptual modeling (Wand & Weber, 2002). The growth of methodological approaches with the lack of theoretical foundations of conceptual modeling became a concern for researchers, and lead to the pejorative term YAMA (i.e., yet another modeling approach). With that in mind, Wand and Weber (2002) proposed a framework for research in conceptual modeling. Their framework is composed of the following elements:

- "*Conceptual-Modeling Grammar* provides a set of constructs and rules that show how to combine the constructs to model real-word domains".

- "*Conceptual-Modeling Method* provides procedures by which a grammar can be used. Usually one major aspect of a method prescribes how to map observations of a domain into a model of the domain".
- "*Conceptual-Modeling Script* is the product of the conceptual-modeling process. Each script is a statement in the language generated by the grammar".
- "*Conceptual-Modeling Context* is the setting in which conceptual modeling occurs and scripts are used" (Wand & Weber, 2002, p.364).

Taking into consideration these four elements, this research proposes a metamodel ontology based on scenarios to work as a guide for building domain ontology (see details in Chapter 6). The metamodel ontology contains constructs (i.e., grammar) to represent the knowledge needed for IS modeling. The research also identifies methods for acquiring knowledge from the domain and mapping to the appropriate constructs. A proposed guideline (i.e., scripts) for using the metamodel to build domain ontologies is offered. Finally, a proof-of-concept prototype to show the feasibility of the scenario metamodel ontology is presented (i.e., context).

Mylopoulos (1992) define Conceptual Modeling as "the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication" (p.3). Modeling results include a conceptual model of the real-world domain of interest and a design model of the information system (Wand & Weber, 1989). The term conceptual model seems to have different interpretations in Information Systems, and is usually used interchangeably with the term design model.

The first step in conceptual modeling activities is the transformation of the perceived real-world system into a model of the system it intends to model. The design model, on the other hand, is the transformation of the conceptual model in the subject world into a model of the information system. In other words, the conceptual model focuses on the problem and the design model focuses on the solution. The view supported in this research regarding the role of conceptual model in systems development is illustrated in Figure 4. Wand and Weber (1989) are concerned that "we model two different domains in systems analysis and design: reality and the information system" (p.82). In this sense, "a conceptual model should reflect knowledge about the application domain rather than about the implementation of the information systems" (Wand *et al.*, 1995, p.285).

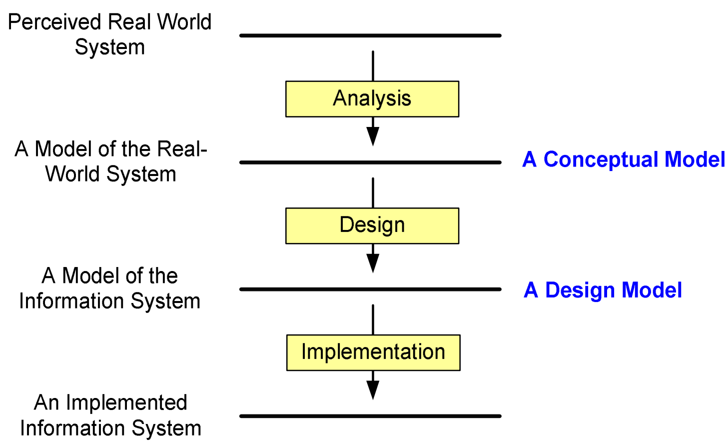

Figure 4: The role of a conceptual model in systems development (from Wand *et al.*, 1995, p.286)

It is important to stress that "in conceptual modeling, there is no 'direct access' to reality" (Wand *et al.*, 1995, p.290). In this case, the conceptual model represents a perception of reality, which means that our view of reality will only be as good as its representation. Since ontology is used to represent the real-world, Wand and Weber (2004) argue that "our description [of the real-world] will only be as good as our ontologies", and because Information Systems are models of real-world systems, they "will only be as good as our ontologies" (Wand & Weber, 2004, p.xii).

According to Fonseca (2007) "to build a [conceptual] model to be used in an information system, we have to start with theoretical informed interpretations of the world as well as objectives for applications so that we may properly select, assess, and organize the facts that make up the system" (p.790). These interpretations (i.e., horizons) are the starting point for defining the ontological commitment with respect to a given domain (Milton & Kazmierczak, 2006; Smith, 2003), that is, the shared view of the domain. In practice, the ontological commitment can be represented as a *metamodel ontology*, also called *top-level ontology*, *frame ontology*, or *upper-level ontology*, containing specific constructs to identify and to describe that shared view. The shared view will inform the creation of conceptual models, and consequently the domain ontology (Fonseca & Martin, 2007). Examples of metamodel ontology used not only for domain modeling but also for IS modeling include OntoClean-DOLCE (Gangemi *et al.*, 2003), BWW Ontology (Wand *et al.*, 1995), and UFO-Unified Foundational Ontology (Guizzardi & Wagner, 2004).

2.2 Ontology

Since its introduction in computer and information science literature in 1967 (Guizzardi, 2005) citing (Mealy, 1967), ontology has become popular and has been used in several domains (Guarino & Welty, 2000; McGuinness, 2001) and for many purposes (Gómez-Pérez *et al.*, 2004). Guarino & Giaretta (1995) presents possible interpretations of the term ontology (Figure 5):

1. Ontology as philosophical discipline
2. Ontology as an informal conceptual system
3. Ontology as a formal semantic account
4. Ontology as a specification of a "conceptualization"
5. Ontology as a representation of a conceptual system via logical theory
   5.1. characterized by specific formal properties
   5.2. characterized only by its specific purpose
6. Ontology as the vocabulary used by a logical theory
7. Ontology as a (meta-level) specification of a logical theory

Figure 5: Possible interpretations of the term "ontology" (from Guarino & Giaretta, 1995, p.25)

As a result of such assortment, the word "ontology" has also received different definitions over the years. The most quoted definition of ontology, according to Gómez-Pérez et al. (2004), is Gruber's definition: "An ontology is an explicit specification of a conceptualization" (p.6). Borst (1997) describes conceptualization as "a structured interpretation of a part of the world that people use to think and communicate about the world" (p.12). For Guizzardi (2005), a conceptualization refers to "abstract entities that only exist in the mind of the user or a community of users of a language" (p.26), and "in order to be documented, communicated and analyzed they must be captured, i.e. represented in terms of some concrete artifacts" (p.26). Guizzardi presents the Ullmann's triangle to illustrate that a language is used to represent the conceptualization of reality (see Figure 6).

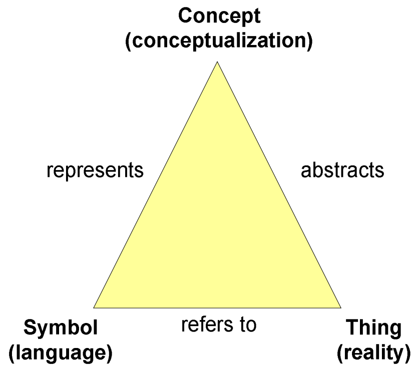

Figure 6: Ullmann's Triangle: the relations between a thing in reality, its conceptualization and a symbolic representation of this conceptualization (from Guizzardi, 2005, p.27)

The definition of ontology adopted for this research follows Studer *et al*. (1998), who state, based on Gruber's (1993) and Borst' (1997)definitions, that "*An ontology is as a formal, explicit specification of a shared conceptualization"* (Studer *et al.*, 1998, p.184) where:

- "A 'Conceptualization' refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon";
- "'Explicit' means that the type of concepts used and the constraints on their use are explicitly defined";
- "'Formal' refers to the fact that the ontology should be machine-readable, which excludes natural language";
- "'Shared' reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group" (Studer *et al.*, 1998, p.184).

In this sense, ontology can be seen as an engineering artifact composed of concepts, attributes, relations and axioms that are used to describe facts (accepted by a community) of a given domain (Guarino, 1998). This artifact is called Computational Ontology (Fonseca & Martin, 2007; Kishore *et al.*, 2004).

Besides the more formal definitions, an ontology may also be given other designations and be seen in different ways (see Figure 7). During the process of screening of publications to find methodologies to build ontologies (see Chapter 4), it was observed that the word ontology was often referred to or defined as (1) a vocabulary of terms about a domain, in cases when the ontology focused on the definition of terms/concepts, and (2) a taxonomy, when the focus was the hierarchical relations (e.g., SubClassOf) between the concepts.

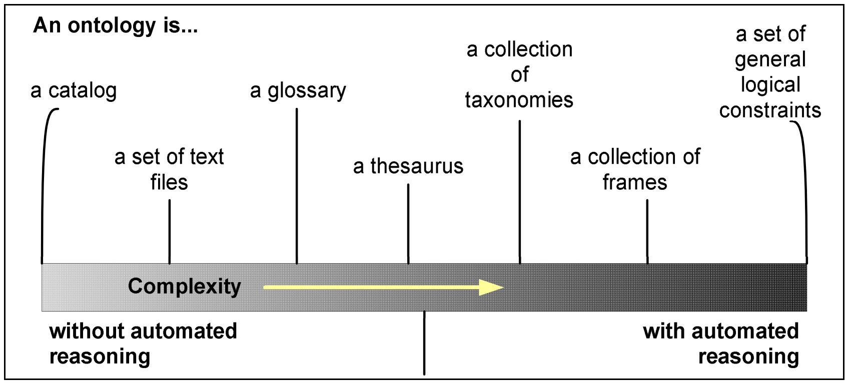

Figure 7: Information artifacts classified as ontology (from Smith & Welty, 2001, p.v)

With regard to the information being represented, ontologies are often referred to as heavyweight and lightweight ontologies. According to Oberle (2006), heavyweight ontologies are highly axiomatized, which allows them "to specify the indented meaning of a vocabulary as precisely as possible" (p.45). Lightweight ontologies, on the contrary, are barely axiomatized, and use mainly concepts and hierarchies to be expressed.

2.2.1 Components of an ontology

The definition of the main components of an ontology has also being presented with some variations. Gómez-Pérez et al. (2004) suggests that the descriptions of the components are influenced by the techniques used to model the ontologies, such as artificial intelligence, software design or database design.

A common view of ontology is as a 4-tuple <**C**, **P**, **R**, **A**> structure that corresponds to the core components of the ontology (i.e., the constructs). A *Concept* (**C**) represents a thing in the real-world (e.g., student). This thing has specific characteristics (e.g., name) that represent its *Properties* (**P**). A thing is associated with other things through *Relations* (**R**). For instance, a student can be enrolled in a course. Finally, both the properties and the relations may need constraints, *Axioms* (**A**), to represent restrictions of the real-world phenomena being represented (e.g., a full time student must be registered for at least 9 credits per semester).

Alternative definitions include *Instance* as a component of the ontology. However, we consider that instances should not be part of an ontology. We agree with Fonseca & Martin (2007) and Stevens *et al*. (2000) in that the combination of instances and the ontology 4-tuple structure creates what is called a knowledge base. Also, Maedche (2002) describes an ontology as a 5-tuple structure, which includes two types of *Relations*: hierarchy (i.e., taxonomic) and function (non-taxonomic) relations. In this research, the core components of an ontology follow the 4-tuple structure described above.

2.2.2 Ontology levels

Ontology can be described in terms of which level of abstraction they represent. Oberle (2006) presents three layers of abstraction for ontologies. First, the generic ontology (also known as Foundational ontology, Top-level ontology, and Upper-level ontology) is domain independent, and corresponds to the basic constructs (e.g., thing, state, process) that can be use to frame many domains. There are a number of top-level ontologies available in the literature (Kishore *et al.*, 2004; Sowa, 2000), such as BWW ontology (Wand & Weber, 1989), Sowa's upper level ontology (Sowa, 2000), UFO-Unified Foundational Ontology (Guizzardi & Wagner, 2004), SUO-Standard Upper Ontology (Niles & Pease, 2001). An example of a top-level ontology is presented in Figure 8, which shows the upper part of the Unified Foundational Ontology (UFO-A). Second, the core ontology is domain dependent, but still at a high level. The core ontology describes the common concepts and associations within a domain. For instance, a core ontology about higher education is pertinent to the education domain, without including the specific of a higher education institution. Third, the Domain ontology is domain dependent, and includes specializations of the core ontology applied to a more specific domain. For example, an ontology about Penn State could be seen as a specialization of a higher education ontology.

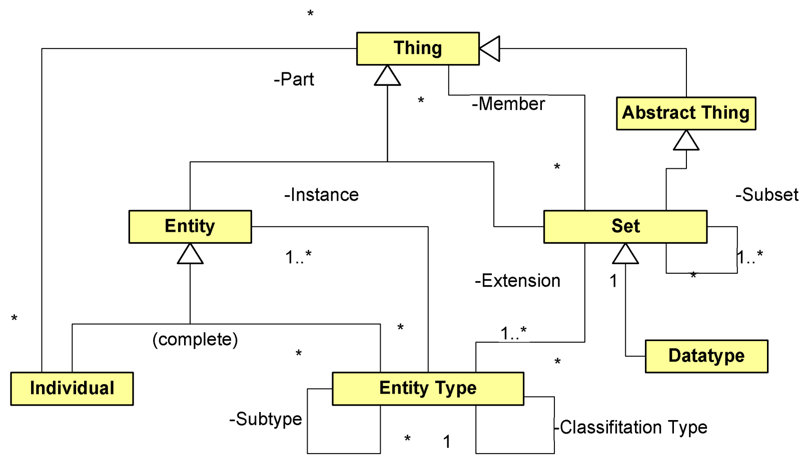

Figure 8: The upper part of UFO-A 0.2 as a MOF/UML model (from Guizzardi & Wagner, 2004, p.351)

Guarino (1998) presents similar classification (see Figure 9). His classification includes a top-level ontology with generic concepts that are independent of the domain, a domain and a task ontologies that are generic to a specific domain (e.g., pharmacy), and an application ontology that contains concepts specialized from the domain and task ontologies. A task ontology describes the concepts and structures for problem-solving (Mizoguchi, 2003), and an application ontology includes concepts related to “the roles played by domain entities while performing a certain activity” (Guarino, 1998, p.9).

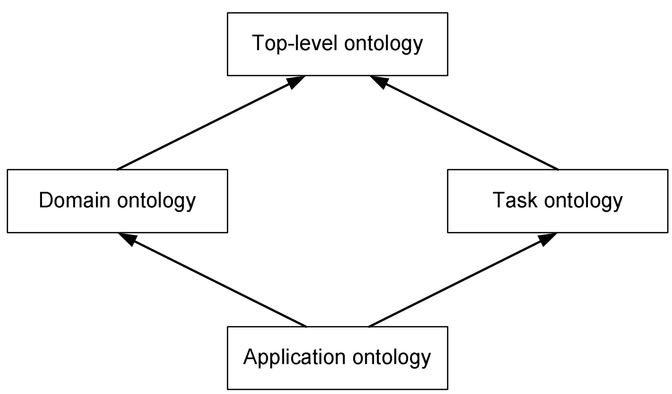

Figure 9: Kinds of ontologies, according to their level of dependence on a particular task or point of view (from Guarino, 1998, p.9)

We see two distinct views of the domain in the process of building ontologies. First, there is a shared vocabulary and meaning of the terms used in the ontology. Second, there is an ontological commitment of how people perceive the domain. For instance, the term *client* in an ontology of a bank has probably a shared meaning among the people involved with the bank (e.g., the employees and the clients themselves). But, when the bank is seen as a system, the term *client* can also represent an *agent* that has specific *goals* when using the services offered by the bank. An agent performs several *tasks* in order to achieve its goals. In this case, the term *client* could be part of a domain ontology of a bank, and the term *agent* could be part of a top-level ontology that makes a commitment to see the bank as a system. This high-level view of the bank as a system, we call a metamodel ontology.

Metamodels "help to further structure, understand, and analyze an ontology" (Davies *et al.*, 2005, p.5). Metamodels can provide a frame for mapping domain concept, and can facilitate the integration of domain ontologies that share the same view of the domain (Davies *et al.*, 2005).

Based on the findings from the analysis of the methodologies to build ontologies synthesized in Chapter 5 and discussed in Chapter 6, we propose a metamodel ontology that is based on scenarios. The underlying components of a scenario provide a domain independent frame to guide the constructions of domain ontologies for IS. With this metamodel, the universe of discourse associated to Information Systems can be viewed as set of events that are combined in order to achieve a goal. The use of scenarios as metamodels is discussed in Chapter 6, where we present a proof-of-concept experiment to build a domain ontology.

2.2.3 Ontology-Driven Information Systems

Guarino (1998) used the term Ontology-Driven Information Systems (ODIS) for systems that make use of formally defined ontologies. According to him an explicit ontology plays a central role in this kind of system thus driving all of its aspects and components. Guarino distinguishes two orthogonal dimensions of ontologies in IS.

First, the <u>temporal dimension</u> which is related to the use of ontologies either at run-time or at development time. Ontologies at *run-time* refer, for example, to ontologies used to facilitate the process of mapping and sharing database schemas and web services structures, or to facilitate the communication between systems. Most of the attention of ontologies in Information Systems seems to be at run time, where ontologies can be distinguished as *ontology-aware IS* (i.e., ontology is available for systems to access its content) and as *ontology-driven IS* (i.e., ontology is part of a system) (Guarino, 1998). Ontologies at *development-time* refer to the process of creating ontologies that describe a given domain, and to the use of these ontologies to support the creation of IS components. Wand & Weber (1989, p.81) propose that “the process of constructing an information system is a transformation from human perceptions to an artifact representing these perceptions” (p.81). On the one hand, designers can make use of ontology as a shared knowledge repository of a specific domain and its related tasks, available as an ontology library (Guarino, 1998). On the other hand, designers can exploit ontology, as a powerful tool to automatically create or to support the creation of IS components (Fonseca, 2007; Guarino, 1998; Kishore *et al.*, 2004).

Second, the <u>structural dimension</u> which is related to the way an ontology can affect the main IS components (Guarino, 1998), either at run time or development time. The main components highlighted by Guarino are:

- The *Information Resources* represent the structure used to store the data of the system (e.g., databases). The most common structure is the Entity-Relationship Model (Chen, 1976). Here, ontology constructs can be mapped to E-R constructs to help generating information resources. Sugumaran & Storey (2006), for example, show the feasibility and usefulness of domain ontologies to support database design.
- The *Application Programs* usually “contains a lot of domain knowledge, which, for various reasons, is not explicitly stored in the database. Some parts of this knowledge are encoded in the static part of the program in the form of type or class declarations, other parts (like for example business rules) are implicitly stored in the (sometimes obscure) procedural part of the program” (Guarino, 1998). The ontology provides knowledge to build application programs, as they reflect the processes that occur in a given domain.
- The *User Interfaces* refer to the inputs and outputs of the communication between the system and the users, and are based on the constraints imposed by the other two components, especially from the application programs. These programs embed information about what, where, when and how the information is needed in a system.

Fonseca (2007) offers a slightly different, but complementary viewpoint of the use of ontologies in Information Systems. He discusses the distinction between the creation and the use of ontologies in IS in terms of the purpose of the ontologies (see Figure 10).

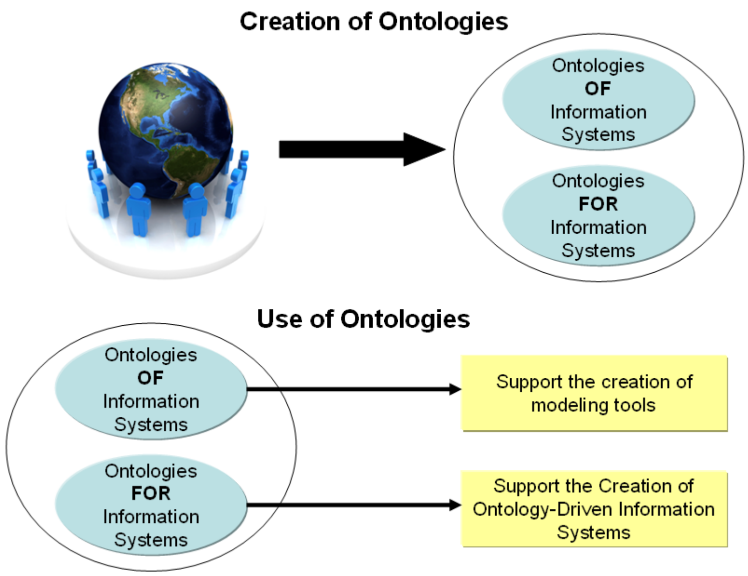

Figure 10: Creation and use of ontologies (from Fonseca, 2007, p.787)

Ontologies are created to describe our view of the world. In the context of ODIS, ontologies *of* Information Systems represent the underpinning theories and structures used to describe the IS domain. These ontologies provide support to the creation of better modeling tools as they become references of how the IS domain is organized. Examples of this type of ontology include the Framework for Information Systems Concepts (FRISCO) ontology (Falkenberg *et al.*, 1998) and the BWW ontology (Wand & Weber, 1989). And, ontologies *for* Information Systems refers to the ontologies created to represent the domain under investigation (i.e., universe of discourse) to which an information system is being designed, such as an ontology describing a gas-station domain. Information Systems, in this case, can be seen as "a human-created representation of a real-world system as perceived by somebody, built to deal with information processing functions in organizations" (Wand & Weber, 1989, p.81). Ontologies about a domain can then be used at development time to support the creation of IS components or at run time to manage the execution of an information system. Guarino (1998, p.10) refers to the ontologies used at run time as ontologies *within* an IS.

This research is particularly interested in the use of ontology at development time of an IS. We envision a two-phase framework (see Figure 1), where the ontology is the backbone of the conceptual modeling. The research focuses on Phase 1 of the framework (see Figure 11), which refers to the construction of domain ontology as an artifact to represent the knowledge of a given universe of discourse for its use in IS modeling. In terms of Fonseca's framework, this research is mainly interested in the creation of ontologies *for* Information Systems.

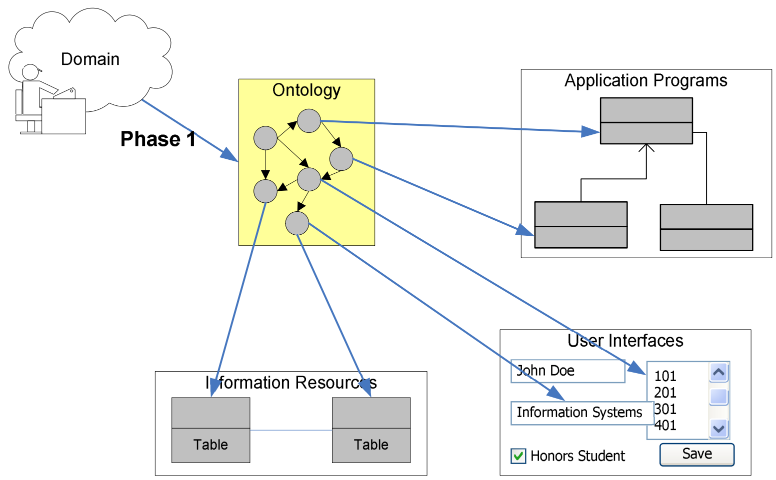

Figure 11: Ontology playing central role in Information Systems

2.2.4 Ontology engineering

Ontology Engineering is "the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies" (Gómez-Pérez *et al.*, 2004, p.5). It includes topics from other disciplines and areas, especially philosophy, computer science, linguistics, knowledge engineering (Devedzic, 2002). In IS, the product of this engineering process is an artifact (i.e., ontology) that describes and represents a given domain (Guarino, 1998). Devedzic (2002) summarizes the topics involved in ontological engineering (see Figure 12).

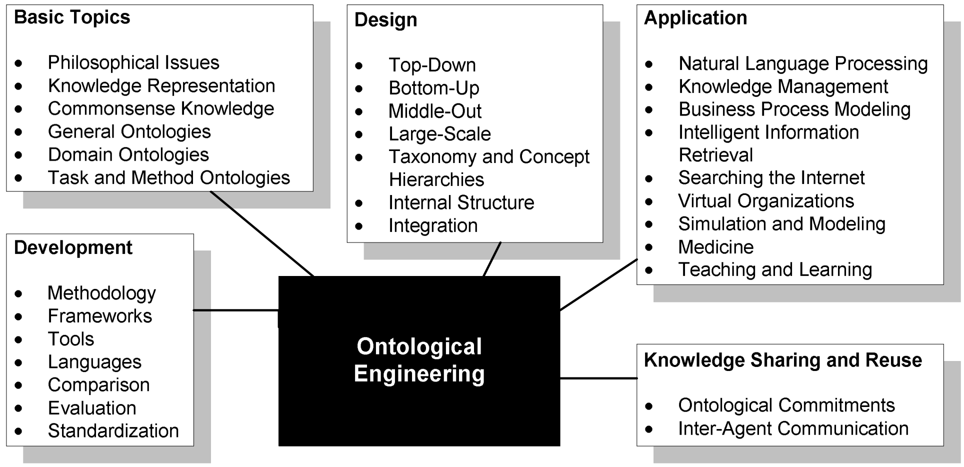

Figure 12: General themes of ontological engineering (from Devedzic, 2002, p.137)

Several methodologies have been proposed to support the process of building ontologies; nevertheless, no methodological approach has been prominent. As described earlier in Chapter 1, a recent survey (Cardoso, 2007) shows that 60% of the participants did not use a methodology to build their ontologies. Because of the lack of methodological principles in the process of building ontologies, ontology engineering is often referred to as a craft activity, rather than an engineering process (Abou-Zeid, 2003; Gómez-Pérez *et al.*, 2004; Jin *et al.*, 2004; Peralta *et al.*, 2005b; Sugumaran & Storey, 2002; van der Vet & Mars, 1998).

There have been a variety of approaches to building ontologies. Husemann & Vossen (2005, p.50) propose a distinction between methodologies that are based on Knowledge Management and Software Engineering approaches. Another categorization of the methodologies takes into consideration the strategy to build the ontologies, that is, building from scratch or building from existing sources (Benslimane *et al.*, 2003; Gómez-Pérez *et al.*, 2004; Pinto & Martins, 2004). In addition, the methodologies can be differentiated in terms of the process to build an ontology (i.e., manual, semi-automatic or automatic approaches).

In this research, we select and investigate 30 methodologies to build ontologies that fall into the category of building ontologies from scratch. Chapter 4 describes the procedures taken to select and to conduct the analysis of the methodologies, and Chapter 5 presents results of the analysis.

2.3 Knowledge representation

Knowledge Representation is one the activities in the area of knowledge engineering that is concerned with the formalization and representation of knowledge about a domain in some machine-readable form (Sowa, 2000). According to Davis *et al*. (1993), knowledge representation plays five fundamental roles (pp.18-27):

- *Role 1: A Knowledge Representation Is a Surrogate*: a representation refers to things that exist in the real world. The descriptions of these things (i.e., tangible and intangible objects, and their relations) with some sort of symbols (e.g., languages) form a model of the world being represented.
- *Role 2: A Knowledge Representation Is a Set of Ontological Commitments*: "all representations are imperfect approximations to reality, each approximation attending to some things and ignoring others, then in selecting any representation, we are in the very same act unavoidably making a set of decisions about how and what to see in the world" (Davis *et al.*, 1993, p.19).
- *Role 3: A Knowledge Representation Is a Fragmentary Theory of Intelligent Reasoning*: the knowledge to be represented for the purpose of intelligent reasoning is based on theories from other fields such as mathematical logic, human behavior, stimulus-response behavior, notion of uncertainty, and utility theory.
- *Role 4: A Knowledge Representation Is a Medium for Efficient Computation*: it refers to the extent in which the representation includes and organizes domain knowledge in a way that it is appropriate for reasoning. This role is concerned with the adequacy and performance of knowledge structure.
- *Role 5: A Knowledge Representation Is a Medium of Human Expression*: it is "a language in which we communicate things about the world" (Davis *et al.*, 1993, p.27).

Knowledge representation is at the core of this research (i.e., Phase 1 of the framework for ODIS: at development time, shown in Figure 1), as we investigate methodologies to build ontologies to represent the knowledge suitable for IS design. Inquiring about "What type of knowledge needs to be represented about an information system?" has already been addressed by Mylopoulus et al. (1990, p.325), who answered the question with:

- "Knowledge about the environment within which the system will function and how the system is expected to interact with that environment";
- "The kind of information the system will be expected to store and the meaning of that information with respect to its intended subject matter";
- "Knowledge about the design and implementation of the information system, which can be used during initial system development as well as during system maintenance";
- "Knowledge about design decisions that led to the particular design/implementation, along with appropriate justifications that relate these decisions to performance or other nonfunctional requirements";
- "Information on the development process itself that led to the system, including the methodology used, the team of developers involved, different system versions, and the like" (Mylopoulos *et al.*, 1990, p.326-327).

In addition, Wand & Weber (1989) state "that a modeling scheme must represent two aspects of a system: structure (statics) and behavior (dynamics)" (p.83). The knowledge involved with the static and dynamic aspects of a system are often referred to as conceptual knowledge (i.e., concepts and relations) and procedural knowledge (i.e., processes and tasks) (Milton, 2007). The conceptual knowledge, also known as declarative knowledge, is commonly described as "knowing that" and the procedural knowledge is described as "knowing how" (Diaper, 1989; Milton, 2007).

There are other types of knowledge besides conceptual and procedural knowledge. Jong & Ferguson-Hessler (1996) add situation knowledge, which includes supplemental knowledge from out of the problem space, and strategic knowledge, which includes the plan of activities to solve a problem. The CommonKADS (Schreiber *et al.*, 2000), a methodology for knowledge engineering and management, describes a knowledge model formed by three types of knowledge: First, domain knowledge describes the static knowledge of a domain, such as concepts and relationships, as well as knowledge about rules and facts. Second, inference knowledge describes how to reason with domain knowledge. Finally, task knowledge describes goals and the activities required to achieve these goals, including task decomposition and task control. CommonKADS includes a Task Model that handles problem solving knowledge (Uschold, 1998). Alavi & Leidner (2001) offer a summary of the different types of knowledge (see Table 1).

Table 1: Knowledge taxonomies and examples (from Alavi & Leidner, 2001, p.113)

Knowledge Types	Definitions	Examples
Tacit	Knowledge is rooted in actions, experience, and involvement in specific context	Best means of dealing with specific customer
Cognitive tacit:	Mental models	Individual's belief on cause-effect relationships
Technical tacit:	Know-how applicable to specific work	Surgery skills
Explicit	Articulated, generalized knowledge	Knowledge of major customers in a region
Individual	Created by and inherent in the individual	Insights gained from completed project
Social	Created by and inherent in collective actions of a group	Norms for inter-group communication
Declarative	Know-about	What drug is appropriate for an illness
Procedural	Know-how	How to administer a particular drug
Causal	Know-why	Understanding why the drug works
Conditional	Know-when	Understand when to prescribe the drug
Relational	Know-with	Understand how the drug interacts with other drugs
Pragmatic	Useful knowledge for an organization	Best practices, business frameworks, project experiences, engineering drawings, market reports

One of the issues raised in Chapter 3 is related to procedural knowledge. As mentioned earlier, procedural knowledge is concerned with the knowledge required to perform some tasks (i.e., know how to do something), specifically the steps needed to complete the tasks. However, as observed in Chapter 5, most of the methodologies investigated do not provide proper support for the identification and representation of this type of knowledge.

2.4 Knowledge acquisition

Knowledge Acquisition "is the requirement to be able to represent domain experts' behavior and thus their knowledge in a form suitable for incorporation into expert systems" (Diaper, 1989, p.23). It is separated into three areas (Milton, 2007): (1) Knowledge Capture: deciding what knowledge to be captured and which techniques to elicit knowledge, (2) Knowledge Analysis: identifying, from the captured knowledge, the important elements and structures to be represented, such as concepts, attributes and relations, and (3) Knowledge Modeling: representing the knowledge elements using some kind of language.

The Welbank's matrix (see Table 2) presents a list of knowledge acquisition methods that can be applied to different types of knowledge (Cordingley, 1989, p.158) citing (Welbank, 1987).

Table 2: Types of knowledge with appropriate knowledge acquisition methods
Key: √, good; ×, bad; ?, difficult, but possible (from Cordingley, 1989, p.158)

	Facts	Conceptual Structure	Causal knowledge	Rules	Weight of evidence	Procedures	Expert's strategy	System's strategy	Context of rules	Explanation	Justification
Interview	√	√	√	?	?		×		?		√
Talking through case-studies	√	×	√	√	×	√	×			√	
Observing interactions							√			√	
Protocol analysis						√	√				
Card-sorting		√									
Multidimensional scaling		√									
Repertory grid		√	×	√	√	×	×				
Induction			×	√							
Task analysis								√			
User interviews								√			
Examining prototype				√	√				√	√	

We consider knowledge acquisition an important activity in knowledge engineering and an essential step towards building ontologies. It should include details to guide the acquisition process. For instance in Table 3, Cordingley (1989) describes the essential information that "should be elicited about any process" (p.170).

Table 3: Process elicitation (from Cordingley, 1989, p.170)

Basic description	*What*: what is done
Temporal ordering	*Before*: what processes come before it in time and have a message or material flow leading to it *Next*: what processes come after it in time and have a message or material flow leading from it *Concurrent*: what processes happen to occur at the same time but which do not share a common "before' or 'next' relationship to it
Contingency information	*Or*: alternative processes; which one is done depends on predetermined control conditions ('OR' processes do not send messages or materials to one another) *And*: all processes are to be done but in any order ('And' processes may or may not send messages to one another)
Establishing hierarchies	*Why*: one is done for the purpose of the other(s); 'Why' relationships establish superordinate/subordinate relationships in hierarchies of purpose; usually the superior sends a control message to the subordinate and receives a data message (a report on progress) back from it *How*: one is done as a means of achieving the other(s); 'How' relationships establish superordinate/subordinate relationships in enabling hierarchies
Production information	*Control*: control messages start and stop processes; they express conditions for actioning processes; identify their source(s) and destination(s) *Concurrent controls*: all messages have to be present and all have to arrive at the same time for the process to be actioned *'or' controls*: if any of the message is present then the process is actioned *'and' controls*: all messages have to be present but they can have arrived in any order for the process to be actioned *Data*: messages which are the information inputs to processes; identify their source(s) and destination(s) and whether they come and go directly or via a store (a 'pool') *Material*: the physical inputs and outputs of processes *Tools*: what is used by people to help them carry out a process; distinguish between types in terms of the process the tool is aiding
Scope information	*Boundaries*: define in terms of the start (successive before?), the end (successive next?), the top level purpose (successive why?), and functional primitives (successive how?) *Who*: the agent, object or processor doing the process *Where*: the physical location of the process, message or material flow *Linked to*: non-functional relationships such as similar to
Evaluate information	*How well*: attainment compared against some goal *How liked*: how (the full range of) target users like doing it *How easy*: whether (the full range of) target users find it easy to do
Ergonomic information	*Health, safety, comfort*: Identify 'hazards'

As shown in Chapter 5, 28 out of 30 methodologies reviewed in this research describe some sort of knowledge acquisition approach as part of their methodological steps. Nevertheless, very few describe details of how and what domain knowledge to acquire.

2.5 Temporal relations

A temporal relation is an approach in knowledge representation involving the events of an application domain, as "they exhibit a history of changes through time" (Mylopoulos *et al.*, 1990, p.7). These changes in the application domain can be described in terms of how events relate to each other. In this case, temporal relations are used to provide the ordering of the events and to document the changes occurring in a domain. According to Sowa (2000) "a change occurs when certain facts that are true in a situation s_1 are no longer true in a later situation s_2" (p.245). Petri nets and flow charts are well known approaches used in object-oriented design and programming to represent changes (Sowa, 2000).

With regard to time, events can be distinguished as a *point of time* or a *time interval* (Helbig, 2006). A *point of time* (also known as an instant or moment) is characterized by an event in which the beginning time (t1) and the end time (t2) are the same (i.e., t1 = t2). An example of this category is to be able to express that someone was born, a credit card was charged or a gunshot was fired at a specific date and time. A *time interval*, on the other hand, denotes an event where the beginning and the end time occur at different points in time (i.e., t1 ≠ t2). For instance, someone can say that a conference will start on Monday at 8:30am with the registration, and will end on Wednesday at 5:30pm with closings from the conference chair. A widespread approach for describing time intervals is Allen's (1983) Interval Algebra. Allen's approach is composed of relations that can be used to describe the temporality between events (see Figure 13).

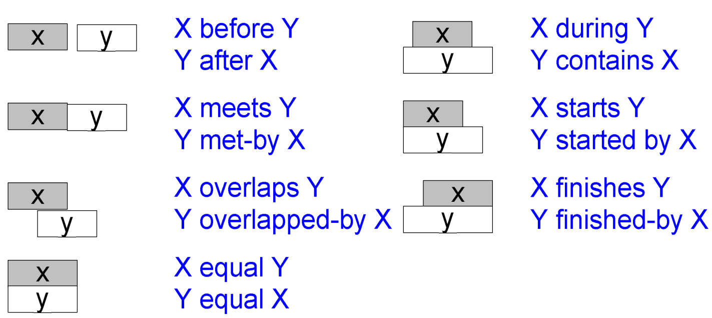

Figure 13: The thirteen temporal relations (adapted from Allen, 1983, pp.835-836)

We see the representation of temporal relations as a crucial part for the future of ODIS. Uschold (2008) envisions ODIS with models that "will not go through the usual steps of code generation, compiling and linking. Rather these models will be already executing as they are being built" (p.14). For that to be achieved, we argue that ontologies should carry more information related to the control and execution of tasks.

Our review of methodologies revealed that only five out of the 30 methodologies provided some support to the identification and representation of temporal relations. Perhaps, as Guarino (1998) explains, "some parts of this [domain] knowledge are encoded in the static part of the program in the form of type or class declarations" (p.13), and are not explicitly represented in the ontologies.

2.6 Scenarios

The frequent use of scenarios in the initial steps of the methodologies to build domain ontologies, reviewed in Chapter 5, brought our attention to investigate scenarios as a key feature for ontology development. In this research, scenarios are being proposed as a metamodel ontology, where the components of a scenario (e.g., agents, goals, and events) can be used as a frame to represent the knowledge of a system, especially with respect to what a system is and how it works.

Scenarios are representations of the real world (Sutcliffe, 2003) with a focus on task interaction and usage (Jarke *et al.*, 1998; Rosson & Carroll, 2002). Scenarios can be expressed with informal (e.g., textual narratives), semi-formal and formal notations (Rolland & Prakash, 2000). We are interested in describing scenarios with the formalizations of ontologies. The use of ontologies to represent domain knowledge is in line with the view that scenario is a story about people and their activities (Alexander, 2004; Johson & Henderson, 2002; Rosson & Carroll, 2001).

Scenarios can also express current and future use of the systems (Weidenhaupt *et al.*, 1998). Carroll (2006) describes *observed* scenarios (i.e., current view) and *envisioned* scenarios (i.e., future view or system to be). To describe scenarios, designers should take into consideration the characteristic elements (see Table 4) of a scenario, which can describe the experience and behavior of actors to achieve their goals (Rosson & Carroll, 2001). Each scenario tells the story of an actor and its relations with other actors and resources, therefore "every scenario involves at least one actor and at least one task goal" (Rosson & Carroll, 2001, p.17).

Table 4: Characteristic elements of user interaction scenarios (from Rosson & Carroll, 2001, p.18)

Scenario Element	Definition
Setting	Situational details that motivate or explain goals, actions, and reactions of the actor(s)
Actors	Human(s) interacting with the computer or other setting elements; personal characteristics relevant to scenario
Task Goals	Effects on the situation that motivate actions carried out by actor(s)
Plans	Mental activity directed at converting a goal into a behavior
Evaluation	Mental activity directed at interpreting features of the situation
Actions	Observable behavior
Events	External actions or reactions produced by the computer or other features of the setting; some of these may be hidden to the actor(s) but important to scenario

The literature about scenarios is vast and with many examples of success (Alexander & Maiden, 2004; Carroll, 1995; Hertzum, 2003; Rosson & Carroll, 2001; Sutcliffe, 2003; Weidenhaupt *et al.*, 1998). Thus, this section is not intended to present an exhaustive description about scenarios. Rather, it focuses on describing how scenarios may become a good fit to the process of developing ontologies. Scenarios are used in the initial steps of ontology development, usually as textual narratives (Gandon, 2002; Grüninger & Fox, 1995). In this research, we suggest using scenarios with a more structured approach, where the components of a scenario become ontological constructs (i.e., metamodel ontology) mapping the elements of a domain ontology (see Chapter 6 for details). In this case, scenarios are part of an ontology, rather than a mechanism to acquire domain knowledge to be represented by an ontology.

Scenarios have been used in software development to represent knowledge about a given domain (Rosson & Carroll, 2002), however most of this knowledge is kept as textual descriptions. For instance, Rolland & Achour (1998) present a description of an automated teller machine (ATM) case study: “The user inserts the user's card in the ATM. The ATM checks if the card is valid. If the card is valid repeat less than four times and until the code is valid; a prompt for code is given by the ATM to the user, the user inputs the code in the ATM, and the ATM checks if the code is valid.” (excerpt from Rolland & Achour, 1998, p.133).

In terms of the semantic continuum (Uschold, 2003), textual descriptions are explicit source of knowledge for humans, nevertheless we also want this knowledge to be used by machines (see Figure 14). We argue that having a metamodel ontology based on scenarios can provide a richer semantic for machines to understand content meaning. Without this layer of understanding on top of the domain ontology (i.e., ontological view of the domain), meaning would be hardwired into the application software (Uschold, 2003). In brief, we propose to represent scenarios with formal ontology to allow both humans and machines to use the content of scenarios.

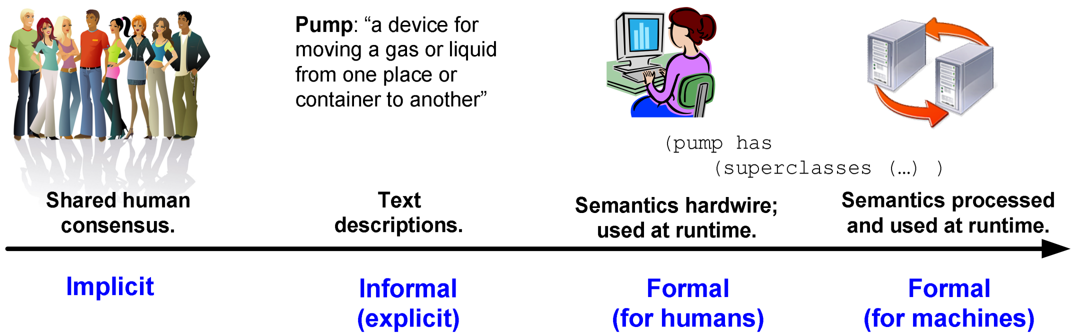

Figure 14: Semantic continuum (from Uschold, 2003, p.28)

Lee (2006) examines the role of scenario use in ontology development. In his analysis of the potential benefits of using a scenario in ontology development lifecycle, he observed that the use of scenarios were most helpful in the following activities of the *Methontology* framework (Gómez-Pérez *et al.*, 2004), namely planning, control, specification, conceptualization and evaluation. Lee's claims with respect to the role of scenario in ontology development are listed in Table 5. The claims reflect his experience working on a project "to develop an ontology that could be used to exchange process descriptions among a wide variety of business process modeling and support systems such as workflow software, flow charting tools, process simulation systems and process repositories" (Lee, 2006, p.273). We support that these claims represent potential benefits of scenarios in ontology development, and we point out that they have similarities with claims from the systems development community (Alexander & Maiden, 2004; Carroll, 2000; Hertzum, 2003; Rosson & Carroll, 2001; Weidenhaupt *et al.*, 1998), where scenarios have been successfully used in the process of systems development, especially in the requirements engineering area (Sutcliffe *et al.*, 1998).

Table 5: Claims of the role of scenario in ontology development (Lee, 2006)

Activity	Claims
Planning	"Because scenarios are concrete descriptions of intended uses, they allow us to more easily envision outcomes and identify early opportunities and risks" (p.274)
	"Because Scenarios focus on a program's uses, they help designers identify stakeholders and set realistic design goals for supporting those stakeholders" (p.274)
	"Scenarios serve as an ideal communication medium among the stakeholders" (p.275)
Control	"Scenarios circumscribe problems, thereby enabling focused use of available resources" (p.275)
Specification	"Scenarios help to identify intended uses and intended users" (p.276)
	"Identification of the intended use establishes the required formality of an ontology while identification of the intended users establishes the required formality of the presentation of an ontology" (p.276)
	"The fact that scenarios focus on uses helps us identify the scope of the ontology to be designed. Scenarios' narrative structure also helps us identify 'the initial set of objects to be represented, its characteristics and granularity' " (p.276)
Conceptualization	"Scenarios' concreteness and focus on uses serves as an ideal communication medium among designers and domain experts" (p.277)
	"A Scenario helps structuring the initial set of domain objects and relations through its ability to naturally accommodate multiple levels of abstraction and incremental formalization" (pp.277-278)
	"A Scenario's narrative structure can be used to generate a systematic method for acquiring and structuring domain knowledge" (p.278)
Evaluation	"Scenarios help to provide bases for concrete testing and evaluations through their concreteness, their focus on use their circumscription of the scope" (p.278)
	"Scenarios provide a basis for evaluating alternative ontologies" (p.278)

2.7 Summary

This chapter described the underlying background of this research in terms of the scope around conceptual modeling, the definition and use of ontologies, the acquisition and representation of domain knowledge, and the use of scenarios as metamodel ontologies guiding the creation of domain ontologies. The next chapter describes the preliminary research on building domain ontologies and the issues (i.e., metamodels, knowledge acquisition, procedural knowledge, and temporal relations) concerning the process of building ontologies for Information Systems. The chapter also discusses the issues found in this process, and reports our need to search for methodologies to build domain ontologies.

Chapter 3.
AN EXPERIMENT IN BULDING ONTOLOGIES FOR IS

Once defined that the focus of our research was to investigate how to build ontologies to be used in IS modeling (i.e., Phase 1 of the framework shown in Figure 1), we decided to conduct an exploratory experiment to build domain ontologies using three methodologies, namely *Methontology* (Gómez-Pérez *et al.*, 2004), *Ontology Development 101* (Noy & McGuinness, 2001), and *Skeletal Methodology* (Uschold & King, 1995). The methodologies were chosen based on the researcher's previous experience in building ontologies with them.

The initial expectation of this exercise was to identify which methodology would provide the most fit ontology to support IS modeling. Nevertheless, in the process of building the domain ontologies with these methodologies we uncovered important issues with respect to metamodel, procedural knowledge, temporal relations and knowledge acquisition. In this section, we introduce the domain data source used in this research, describe the steps of the three methodologies used to build our domain ontology, discuss the issues found that are important for IS modeling and ontology development, and describe the need for searching other methodologies to build ontologies.

3.1 Domain data source

The domain under investigation refers to the process of adopting a cat from the local animal shelter called Centre County PAWS (henceforth PAWS). The typical life-cycle of a cat at PAWS proceeds from the time a person brings a cat to the shelter until the time the cat leaves the shelter to the adopter's home. To achieve this goal, several activities have to be performed, such as evaluating the health of the cat, posting information about the cat on the website, approving and relocating the cat to foster homes until adoption, approving adopters, releasing the cat, and publishing the cat's happy end story. The data used for this experiment is secondary data available online at ucs.ist.psu.edu. The material is from a case study of the book *Usability Engineering:*

Scenario-Based Development of Human-Computer Interaction (Rosson & Carroll, 2001). The ontology created for this domain is called *PAWS Ontology*.

3.2 Methodologies used to build the domain ontology

The main goal with this experiment was to build an ontology about the PAWS domain that would represent the domain knowledge to be used in IS modeling. Following, we present a summary of the steps recommended by each methodology (i.e., Methontology, Ontology Development 101, and Skeletal Methodology) in building the domain ontology.

3.2.1 Methontology

This methodology includes eleven tasks (Gómez-Pérez *et al.*, 2004, pp.132-140):

- *Build glossary of terms*, including names, synonyms, acronyms, description and type of term (i.e., concept, attribute, relation, or instance).
- *Build concept taxonomies*. This task organizes the terms as hierarchical categories represented by a *Subclass-Of* relation.
- *Build ad hoc binary relation diagrams*. The objective of this task is to identify relationships between concepts from the same (or different) taxonomy.
- *Build concept dictionary*. This includes for each concept a list of their class attributes, instance attributes, and relationships.
- *Describe ad hoc binary relations*. The goal is to describe each relationship in details, which includes relation name, source concept, source cardinality, target concept, mathematical properties, and inverse relations.
- *Describe instance attributes*, including instance attribute name, concept name, value type, measurement unit, precision, range of values and cardinality.
- *Describe class attributes*, including attribute name, value type, measurement unit, precision, cardinality, and values.
- *Describe constants*. This task defines the constants identified in the glossary of terms. Each constant is described in terms of name, value type, value, and measurement unit.

- *Describe formal axioms*. Each axiom is defined using first-order logic.
- *Describe rules*. This task is similar to the previous task, but with description based on the IF-THEN statements.
- *Describe instances*. According to the authors, this task is optional.

3.2.2 Ontology development 101

The authors of the methodology suggest an interactive process composed of seven steps (Noy & McGuinness, 2001, pp.5-12):

- *Determine the domain and scope of the ontology*. For the domain, one should answer questions such as what is the domain being covered by the ontology? What is the audience for the ontology? For the scope, one should elaborate competency questions that "the ontology should be able to answer" (p.5). The competency questions can help limit the scope and content of the ontology as the ontology should have enough information to answer the questions;
- *Consider reusing existing ontologies*. Several ontologies have been made in some machine-readable form and are available through ontology libraries on the internet;
- *Enumerate important terms in the ontology*, including their meaning and properties;
- *Define the classes and the class hierarchy*. In this step, one can select a specific class and adopt a top-down, a bottom-up, or a combination of both approaches to define the hierarchical arrangement of the classes;
- *Define the properties of classes-slots*. This step will use the list of terms created previously. The properties can be either a data-property or an object-property (i.e., relationships with another class);
- *Define the facets of the slots*, such as cardinality, value-type (e.g., String, Number, Boolean, etc.), domain and range;
- *Create instances*.

3.2.3 Skeletal methodology

The methodology is the report of the authors' experience in building ontologies. It includes the following stages (Uschold & King, 1995, pp.2-4):

- *Identify Purpose* is the process of identifying the scope of the ontology and its intended uses. Competency questions could be used to help identifying the purpose and narrowing the scope.
- *Building the Ontology* is divided into three sub-stages*:*
    - *Ontology Capture* refers to the identification of concepts and relationships with unambiguous definitions.
    - *Ontology Coding* refers to the explicit representation through a formal language, and consequently, a commitment to the meta-ontology of the language.
    - *Integrating Existing Ontologies*.
- *Evaluation* is the verification of the ontology with relation to a frame of reference, such as competency questions, that is, if the ontology is able to answer the competency questions or if it is in accordance to the initial requirements specifications.
- *Documentation* can be seen as guidelines for naming conventions and other practices, as well as a complementary description of the rationale for building the ontology.

3.3 Issues found in the process of building the domain ontology

The methodologies offered fairly straightforward steps to build the ontology. Nevertheless, we argue that building domain ontologies is far from a trivial activity, and requires some thought from the designer. For instance, all three methodologies contain a step referring to the identification of concepts in the domain under investigation, however little guidance is offered in terms of how to properly identify these concepts and their relationships.

Our preliminary investigation of the process for creating the ontology, and the content created revealed four issues that we consider important to the development of ontologies to be used in IS modeling: metamodel, procedural knowledge, temporal relations, and knowledge acquisition. Figure 15 presents an excerpt from the domain ontology that will be used to discuss the issues.

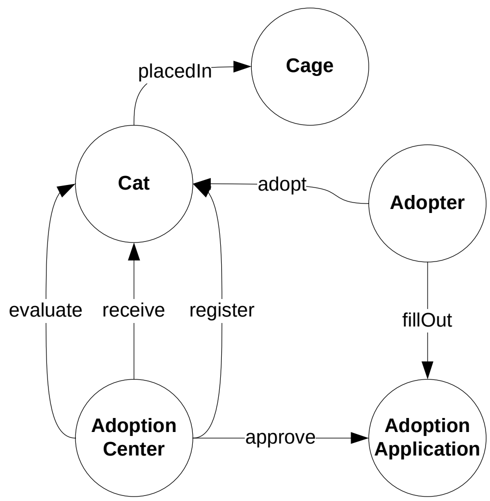

Figure 15: Excerpt from the domain ontology

3.3.1 Metamodel

Metamodels can provide a frame for mapping domain concept, and can facilitate the integration of domain ontologies that share the same view of the domain (Davies *et al.*, 2005). In Figure 15, the concepts Cat (i.e., the animal for adoption) and the concept Cage (i.e., a place to enclose the pet at the animal shelter facility) do not have a clear distinction between them. Both concepts are represented simply as two different concepts connected by a relationship. Although we acknowledge that the representation is correct, we would like to point out that it is also incomplete in terms of identifying the proper roles of the concepts. People should be able to understand these two concepts and the relation between them. However, computers would require some extra information to support the semantics of that relation.

We suggest the use of a metamodel ontology to provide a refined understanding of the domain and a frame linking the metamodel ontology with the domain ontology. A metamodel ontology defines the ontological commitment, that is, how we perceive the real world phenomena (Kurtev, 2007). Thus, we envision a mapping from the metamodel ontology to the domain ontology to increase the understanding of the domain and to facilitate its interpretation. For instance, the generic concept of an agent in the metamodel ontology is mapped to the concept of an adopter in the domain ontology, which can then point to an instance of an adopter called Joe (see Figure 16). By connecting the concept agent with the concept adopter, we see an enhancement on our understanding about an adopter. For instance, an adopter, being an agent, can perform actions and trigger events that could change the state of other concepts.

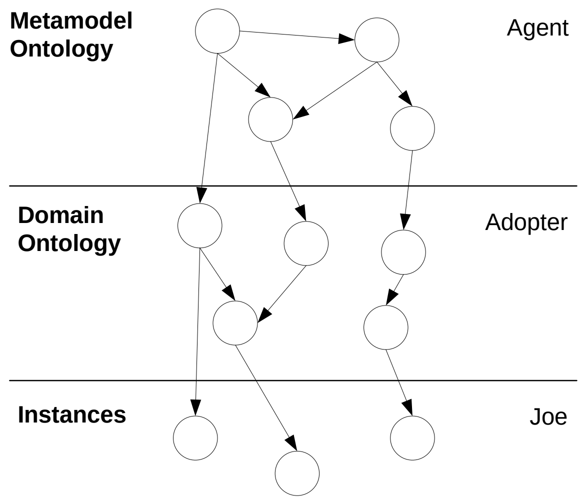

Figure 16: Metamodel and domain level ontologies

3.3.2 Procedural knowledge

Procedural knowledge refers to a sequence of tasks needed to achieve a goal, that is, the description of how to do things (Milton, 2007). It refers to the knowledge “generated whenever people refine step-by-step processes for standardizing simple, everyday work processes” (Allee, 1997, p.126).

In the PAWS domain the high-level goal is to get a cat adopted. This goal depends on the achievement of several sub-goals and tasks. As shown in Figure 15, it is possible to identify at least three distinct clusters of activities that should be performed to achieve the overall goal. First, when someone brings a homeless cat to the animal shelter, second, when a potential person (called adopter) applies for adopting a cat, and finally, when a match between adopter and cat occurs and the process of adoption is approved.

An ontology that will represent the PAWS domain and will be the basis for the design of an information system for the same domain should include a proper representation of the tasks above. These tasks portray the main activities in the domain, and can be used to understand what the system is and how it works.

Procedural knowledge is an important feature for Information Systems. However, it has not been covered by the three methodologies used in this experiment. Also, as discussed in Chapter 5, only 11 out of the 30 methodologies analyzed included some support to representing procedural knowledge.

3.3.3 Temporal relations

A temporal relation is an approach in knowledge representation involving the events of an application domain, as “they exhibit a history of changes through time” (Mylopoulos *et al.*, 1990, p.7). In Figure 15, the tasks are represented with relation to the concepts with no particular temporal identification, which makes almost impossible to define the chronological order of the tasks. Assessing the correct order of the tasks can provide information about the timeline for performing the tasks, which tasks are needed to achieve goals, and how the tasks depend on each other. We cannot identify which task comes first or later in the process of adopting a cat, or say whether the tasks belong to the same cluster of activities (i.e., scenarios). For instance, in the overall process of adopting a cat, a person receiving a cat at the shelter is the first task and an adopter adopting a cat is the last task. However, the temporality between these tasks in Figure 15 is non existent.

3.3.4 Knowledge acquisition

Knowledge Acquisition refers to the process of capturing and representing domain knowledge (Diaper, 1989). None of the methodologies used in this preliminary experiment provide a systematic method for identifying and capturing knowledge suitable for IS modeling, especially with regards to procedural knowledge and temporal relations. The lack of appropriate guidelines puts pressure on domain experts and ontology designers, who have to find on their own, ways to identify the relevant knowledge to be represented by the ontologies.

Both the Ontology Development 101 and the Skeletal Methodology suggested the use of competency questions as a primary resource for capturing domain knowledge. Grüninger & Fox (1994) consider that competency questions work as requirements that the ontology created should be able to answer. In terms of identifying concepts, the Skeletal Methodology suggests three strategies: top-down (i.e., most generic concepts first, then specialize the concepts), bottom-up (i.e., specialized concepts first, then generalize the concepts), or middle-out (i.e., basic concepts first, then specialization or generalization of the concepts). Uschold & King (1995) reported their use of a brainstorm technique to identify relevant concepts. The Methontology and the Ontology Development 101 also suggests the ontologist identify relevant terms in the domain to build a glossary of terms. The glossary is the basis for adding the properties of concepts and the relationships between the concepts (i.e., taxonomic and non-taxonomic relationships).

3.4 Searching for methodologies to overcome the issues

The main reasons motivating our search for methodologies to build ontologies are: (1) the four issues (i.e., metamodel, procedural knowledge, temporal relations, and knowledge acquisition) in the process of building ontologies for IS modeling, (2) 60% of the participants in Cardoso's (2007) survey did not use any methodology to build ontologies, (3) three out of nine methodologies shown in Cardoso's survey presented shortcomings with respect to the issues found, and (4) the process of building ontologies is still considered an ad-hoc activity (Abou-Zeid, 2003; Gómez-Pérez *et al.*, 2004; Jin *et al.*, 2004; Peralta *et al.*, 2005b; Sugumaran & Storey, 2002; van der Vet & Mars, 1998). This research aims to identify methodologies that can overcome these issues, as well as, to investigate sound methodological principles used in the building of ontologies for IS.

3.5 Summary

This chapter described the preliminary research of building ontologies using three methodologies. The process of building the ontology and the ontology itself revealed four important issues that should be taken into consideration for the design of ontologies for Information Systems. The issues are related to metamodels, knowledge acquisition, procedural knowledge and temporal relations.

In the next chapter, we describe the search for methodologies that can overcome the issues discussed above and provide specific methods for building ontologies for IS.

Chapter 4.
LITERATURE REVIEW OF METHODOLOGIES

This chapter discusses the research design and methods used in this research. The chapter starts with a brief overview of the Systematic Reviews approach and its research process. The remainder of the chapter describes the steps taken in the process of conducting the review. The chapter concludes with a presentation of the methodologies found and a description of the criteria used to analyze them.

4.1 Systematic reviews

Systematic Reviews is a research methodology that provides a scientific support to the processing of large bodies of information on a specific topic (Petticrew & Roberts, 2006). Also known as research synthesis (Cooper & Hedges, 2009), Systematic Reviews aims to produce high quality literature review that follows specific guidelines to find and to analyze primary studies. This approach makes the research process clear, and less prone to bias. Explicitly stating the steps taken and the rationale for making decisions throughout the review process, allows for a more rigorous, and reliable literature review (Torgerson, 2003). According to Torgerson, the aims of a Systematic Review are:

- "to address a specific (well-focused, relevant) question";
- "to search for, locate and collate the results of the research in a systematic way";
- "to reduce bias at all stages of the review (publication, selection and other forms of bias)";
- "to appraise the quality of the research in the light of the research question";
- "to synthesize the results of the review in a explicit way";
- "to make the knowledge based more accessible";
- "to identify gaps; to place new proposals in the context of existing knowledge";
- "to propose a future research agenda; to make recommendations";

- "to present all stages of the review in the final report to enable critical appraisal and replication" (Torgerson, 2003, p.7).

There are several guidelines to conduct Systematic Reviews. Kitchenham (2004) presents a summary of the main phases involved in a systematic review (Figure 17).

Phase I: Planning the review

Stage 1: Identification of the need for a review

Stage 2: Development of a review protocol

Phase II: Conducting the Review

Stage 1: Identification of research

Stage 2: Selection of primary studies

Stage 3: Study quality assessment

Stage 4: Data extraction & monitoring

Stage 5: Data synthesis

Phase III: Reporting the review

Figure 17: Stages in a systematic review (adapted from Kitchenham, 2004, p.3)

The following sections of this chapter describe the stages of Phases I and II. Stage 5 (Data synthesis) will be presented in Chapter 5. Phase III (reporting the review) refers to the dissemination of the review "to ensure that readers are able to properly evaluate the rigor and validity of a systematic review" (Kitchenham, 2004, p.22). Together, Chapter 4 and Chapter 5 report details of this review.

4.2 Identification of the need for a review

Many researchers have claimed that the process for building ontologies is more a product of a craft activity rather than a engineering activity (Abou-Zeid, 2003; Dong-Soon *et al.*, 2007; Peralta *et al.*, 2005a; Qing *et al.*, 2007). Cardoso's (2007) survey shows empirical evidence to support these claims, where 60% of the participants did not use any methodology to construct ontologies (see Figure 2 on page 3). The remaining participants are distributed among nine methodologies (i.e., On-to-knowledge methodology, METHONTOLOGY, Uschold and King's method, Cyc method, Grüninger and Fox's methodology, DILIGENT method, KACTUS method, SENSU method, Noy and McGuinnness method) and a broad category "other methodologies".

As described in Chapter 1, the goal of this dissertation is to identify methodological guidelines to build ontologies suitable to IS design. A thorough analysis of the methodologies for building ontologies should uncover important lessons learned and practical approaches that can support the process of building ontologies for the purpose of modeling and designing Information Systems. It should also provide a list of issues that still need to be addressed to allow that to happen. With that in mind, this research reviewed the literature "to comprehensively locate and synthesize research that bears on a particular question, using organized, transparent, and replicable procedures at each step in the process" (Littell *et al.*, 2008, p.1). Moreover, a Systematic Review is needed to "provide an authoritative overview of the current evidence, and suggest directions for future research" (Petticrew & Roberts, 2006, p.28).

4.3 Development of a review protocol

Before conducting the review, a researcher must explicitly describe the review protocol, which "is an *a priori* statement of the aims and methods of the review" (Torgerson, 2003, p.26). The protocol is intended to reduce the researcher bias and to provide a guideline for conducting the research that supports all decisions incurred in the review process. In this case, a protocol would help researchers to avoid being influenced by individual studies or by their expectations (Kitchenham, 2004; Torgerson, 2003).

A protocol includes information about the background context of the review, the research questions to be answered, the search strategy to find primary studies, the inclusion and exclusion criteria to select a study, the procedure to check the quality of the studies, the strategy to extract data from the studies, and the strategy to perform the synthesis of selected studies (Kitchenham, 2004). Next, we describe the protocol used to conduct a systematic review of methodologies to build ontologies.

4.3.1 Background

The review targeted publications regarding methodologies to build ontologies for Information Systems. It includes publications from major bibliographic databases that cover the areas of Information Systems, Information Science and Computer Science. It also considers publications that introduce and propose new methodologies. Investigating these methodologies should provide valuable information about how ontologies are created and the main issues that need to be tackled.

The research on Systematic Reviews presents its own vocabulary, which are used throughout this chapter. Following, we explain how to interpret the main terms used in this review:

- *Study* is used to indicate a research describing the design of a specific methodology to build ontologies;
- "Individual studies contributing to a systematic review are called *primary* studies; a systematic review is a form a *secondary* study" (Kitchenham, 2004, p.1);
- *Methodologies* will be mainly used as an allusion to the pool of methodologies identified and selected for this review;
- A *methodology* can have one or more *primary* studies related to it;
- *Citation* refers to the description (e.g., title, authors, abstract, pages, etc.) of a publication within an electronic database;
- *Paper*, *published material*, *document*, and *report* can be used interchangeably to refer to the full text (usually as a pdf file) of a publication;
- *Survey paper* refers to a paper that discusses (e.g., compare, summarize, analyze) existing methodologies. Survey papers are used as alternative mechanisms to help identifying other relevant methodologies that have not been found by the search of the bibliographic databases.

4.3.2 Research questions

Kitchenham (2004) suggests that "the most important activity during protocol is to formulate the research question" (p.5), because the entire review is formulated with the intention to answer the research questions. In other words, the types of questions asked will influence and determine the procedures to conduct the review. Petticrew & Roberts (2006) warn to "*never start a systematic review until a clear question (or clear questions) can be framed*" (p.35).

The major question that drives this dissertation is "*How can we build ontologies that are appropriate to IS modeling?*". In order to build such ontologies, we have to investigate the existing methodologies to build ontologies, which leads to the following review question:

- *Which methodologies are effective to build ontologies for the purpose of IS design?*

Note that we have two distinct research questions, first is the main research question of this dissertation, and second is the Systematic Reviews question that is guiding the review of methodologies. This review question is relevant to anyone planning to represent the knowledge of a given domain with the purpose of designing Information Systems. However, it is a high level question that is already biased towards the pre-conception of the existence of a satisfactory methodology. On the one hand, the review could find one or more methodologies capable of building ontologies for IS design, and the answer to the question would be a list of potential methodological steps. On the other hand, the review could end up not finding any satisfactory methodology and the answer to the question would not produce useful results to practitioners.

To avoid biases, the main question is broken down into sub-questions that should account for the individual contribution of each methodology towards the process of building ontologies for Information Systems, independently of its overall effectiveness as a complete methodology. The new review questions are framed around the issues discussed in Chapter 3 (i.e., metamodels, procedural knowledge, temporal relations, and knowledge acquisition), as follows:

- *How does the methodology support the use of metamodels to guide the construction of domain ontologies?*
- *How does the methodology support the representation of procedural knowledge to describe the tasks performed for achieving goals?*

- *How does the methodology support the representation of temporal relations of the tasks?*
- *How does the methodology support a systematic approach for acquiring domain knowledge?*

These questions should help the review to produce meaningful results to practitioners and researchers by uncovering specific practices to enhance the construction of ontologies for the purpose of modeling Information Systems, and by identifying new research directions to improve existing methodologies.

4.3.3 Search strategy

The review focused on Ontology Engineering methodology and its application to Information Systems. Therefore, the areas of interest covered by the search of published materials are Information Systems, Information Sciences and Computer Science.

Following are the bibliographic databases selected as sources of publications:

- ACM Digital Library
- IEEE Xplore
- SpringerLink
- Elsevier ScienceDirect
- Web of Science
- Proquest

Finding “methodologies to build ontologies” is the main topic behind the search, and the main drive to construct the query used to search the electronic databases. By manually parsing the topic above, three keywords are extracted: “methodologies”, “build”, and “ontologies”. Each keyword is further analyzed to produce the final list of keywords. First, the equivalent singular word for methodologies (i.e., methodology) and ontologies (i.e., ontology) are included on the list of keywords. Second, synonyms for the word “build” (i.e., design, construct, develop, and create) are included as well. Finally, the word “method” is included as it can refer to a “*orderly process or procedure used in the engineering of a product or performing a service*” (IEEE, 1990) cited in (Gómez-Pérez *et al.*, 2004, p.108).

As shown in Figure 18, a methodology is composed of methods. In this case the topic “methods to build ontologies” is equally relevant to the search of published materials. The word “technique”, however, has a broader meaning and refers to the application of methods (Gómez-Pérez *et al.*, 2004). Therefore, it will not be included on the list of keywords.

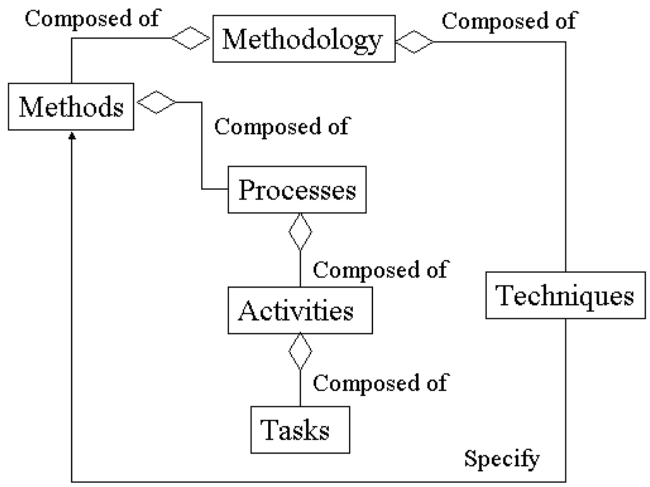

Figure 18: Graphical representation of terminological relationships in methodologies (extracted from Gómez-Pérez *et al.*, 2004, p.109)

The next step is to combine the keywords to formulate the query that will be used as input for the search. Figure 19 shows the final list of keywords. The query can be formulated by combining one word from each box and using the Boolean operator “AND” to concatenate them. For instance, “methodology AND build AND ontology”, or “methodologies AND create AND ontologies”.

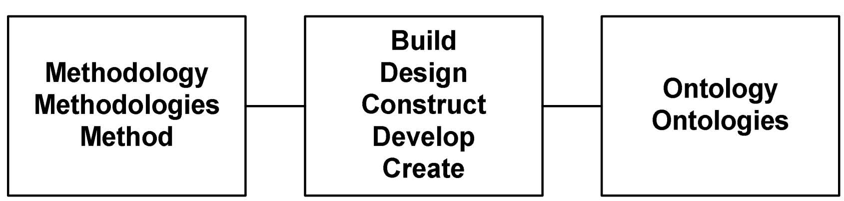

Figure 19: List of relevant keywords for performing the search

With the keywords defined, it is now time to build the query to be used in the search of published materials. However, there are many possible combinations to manage. To overcome this problem, the keywords from each box are enclosed within parentheses and the Boolean operator “OR” is included between the keywords. This means that any of the keywords can be selected. For instance, "(methodology OR methodologies OR method)".

Alternatively, the use of the wildcard asterisk (*) can be used to allow variations of the same root word. For instance, instead of using the keywords "methodology", "methodologies", and "method", the term "method*" would include all of them and any other word starting with "method". However, during the preliminary tests with the query, it was observed that querying the SpringerLink database using wildcards produced no results. Querying the remaining bibliographic databases, however, would produce a large number of citations, which would make the screening of publications unfeasible considering the resources and time available. Hence, the final query (see Figure 20) used to search the bibliographic databases does not contain wildcards.

(methodology OR methodologies OR method) AND
(build OR design OR construct OR develop OR create) AND
(ontology OR ontologies)

Figure 20: Query used to search the electronic databases

With the intent to avoid a high recall of citations, the query was used only with the "title" and the "abstract". The rationale for such criterion is that if the keywords are found in such a restrict space; the publication would more likely be a candidate study.

Finally, only publications written in English were considered, and the year of the publication was not a restriction for searching the databases.

4.3.4 Inclusion and exclusion criteria

The criteria to reject or to select a primary study should be clearly specified in the protocol (Kitchenham, 2004). The list of criteria can be used as a checklist to help the researcher to make a decision whether the primary study fits the purpose of the review.

There will be two phases in the process of selecting primary studies. Phase 1 refers to the selection of studies by applying the selection criteria against their titles and abstracts. If a document passes the inclusion criteria but it is not clear about the exclusion criteria, it will be set aside for further investigation because of the lack of arguments to make a decision. That means the researcher needs to take into consideration both the reasons to include and the reasons to exclude a document from the pool. For a citation to be considered relevant to the review all inclusion criteria must be met and none of the exclusion criteria should be met. Phase 2 refers to the process of retrieving, for each selected citation, the papers from the electronic databases for a closer look at their content. In this case, the selection criteria are applied again to the body of the document.

The inclusion criteria are:

- Publications that propose a methodology to build ontologies;
- Clear description of the methodological steps;
- Methodologies to build ontologies from scratch;
- Documents written in English.

The exclusion criteria are:

- Publications that do not propose a methodology to build ontologies;
- Primary studies with missing or unclear description of the methodological steps to build ontologies;
- Documents not written in English;
- Position papers or tutorials;
- Methodology that is domain dependent (e.g., Medical, Health, Chemistry, etc.);
- Methodologies that build ontologies by extracting/learning information from texts, documents, corpus, etc.;
- Methodology for automatic or semi-automatic construction of ontology;
- Methodologies to build ontologies by merging, integrating, matching or reusing existing ontologies;
- Methodologies that use ontology (Ontology-based) rather than methodologies to build ontologies, including methodologies for MAS (Multi-Agent Systems);
- Methodologies for collaborative or distributed construction of ontologies focusing on collaboration rather than on ontology construction.

We are interested in the construction of the very first ontology rather than reusing existing ontologies to build new ones. The construction of the ontology should focus on the interaction of domain experts and ontology designers building a domain ontology, rather than the use of some automatic mechanism for data extraction from documents. Also, the paper should mainly describe a methodology to build ontology, rather than a methodology that uses ontology as part of its approach, or that just uses an existing methodology.

4.3.5 Quality assessment

Assessing the quality of a primary studies can have different interpretations (Littell *et al.*, 2008; Petticrew & Roberts, 2006). On the one hand, it can relate to the evaluation of the included studies in terms of its methodological approaches, research design, and research results. Nonetheless, this approach can also be a source of bias as researchers could be evaluating either the quality of the study or the quality of the report describing the study (Torgerson, 2003).

The limitations on the size of reports, imposed by most of publication venues, could be a factor threatening the credibility of a study. For instance, the same research published as a conference paper, a journal paper or a PhD dissertation would present various degrees of in-depth discussion of process for conducting the research. On the other hand, the quality of the study can be evaluated in terms of its support to answering the review questions, that is, how the research fits and contributes to the purpose of the review, rather than how it was conducted or reported. In both cases, appraisal questions can be used "to aid informed judgments […] to quality assessment" (Petticrew & Roberts, 2006, p.152), where the questions can act like a checklist that would determine if a study contains information that is relevant to the review process.

The review proposed here follows the later interpretation and uses appraisal questions to evaluate the individual contribution to the review as a measure of relevance of the included studies. For the purpose of this review, a good study should include (if possible): detailed information about the steps of the methodology to build ontologies, and details of how to identify and to describe the main components of an ontology, such as concepts, properties, relationships and axioms; a description of how to acquire relevant domain knowledge; a description of how the procedural knowledge is acquired and represented and how dependencies and temporal relations are described; examples of how to use the methodology and how to apply the steps; a list of the existing methodologies used to identify issues that influenced the design of the methodology; and a description of how approaches from existing methodologies are adopted. Table 6 presents the appraisal questions formulated to evaluate the studies.

Table 6: Appraisal questions for study quality assessment

#	Question
1.	How well is the process of acquiring domain knowledge described?
2.	How well is the process of identifying concepts and properties described?
3.	How well is the process of identifying relationships described?
4.	How well is the process of representing procedural knowledge described?
5.	How well are the task dependencies and temporal relations described?
6.	How well is the process of identifying axioms described?
7.	How well is the use of ontology levels described?
8.	How well is the process of mapping the ontology levels described?
9.	How well are the steps of the methodology described?
10.	How clear are the examples of applying the methodology?

4.3.6 Data extraction strategy

Each electronic database was searched with the query presented in Figure 20. The resulting citations were exported with a format compatible with a reference manager software such as EndNote, Reference Manager, ProCite, and RefWorks. SpringerLink, IEEE Xplore and Elsevier ScienceDirect use the RIS (Research Information Systems) file format, Web of science uses the ISI Common Export Format, Proquest exports as a text file, and ACM Digital Library can be exported directly into EndNote.

All exported citations included an abstract, which was used in Phase 1 of the search (see Section 4.3.4) to evaluate the inclusion and exclusion criteria. The exported files were imported into EndNote, where each database source had its own library. The citations selected for Phase 2 (see Section 4.3.4) were added into an Excel spreadsheet with information about the year of publication, database source, authors, document title, and type of publication (i.e., journal, conference, technical report, and dissertation). Duplicated citations (i.e., same title, author and abstract) from different sources were combined, and the database source column of the first occurrence was updated to include the name of the new database source. For each citation listed, a copy of the document was retrieved to perform a second round of evaluation of the inclusion and exclusion criteria.

The issues discussed in Chapter 2 (i.e., metamodels, knowledge acquisition, procedural knowledge, and temporal relations) and some criteria developed during the design of the review protocol were used to frame the coding system.

For each study selected, data related to the following coding categories was identified and extracted for synthesis:

1. **Knowledge Acquisition**: this criterion aims to identify methods that can help in the process of acquiring knowledge about a given domain;
2. **Identify Concepts**: shows how a methodology supports the identification of domain concepts and their related properties;
3. **Identify Relationships**: shows how a methodology supports the identification of the relationships between concepts;
4. **Identify Tasks**: this criterion covers how the methodologies identify and represent the procedural knowledge need to achieve goals;
5. **Identify Temporal Relations**: refers to particular ways used to identify and to represent the chronology and dependencies of the tasks within the ontology;
6. **Identify Axioms**: an important feature of ontology is the possibility of representing relevant constraints of the domain. This criterion should provide valuable information on how the methodologies propose the identification and description of theses constraints as well as what logic approaches are used to describe the constraints (e.g., first-order logic);
7. **Ontology Levels**: developing ontologies with the help of a metamodel ontology can provide additional knowledge about the domain. This criterion focuses on the methodologies that are using different levels of ontology;
8. **Mapping**: if a methodology has adopted different levels of ontologies, it is important to know if guidelines for identifying the constructs of the higher-level ontologies and for mapping the levels have been proposed;
9. **Methodological Steps**: describes the sequential steps proposed to build an ontology;
10. **Examples**: identifies the specific examples used to illustrate how to apply the methodology or some of its steps to build ontologies;

Motivated by Guarino & Welty's (2000) caveat about the lack of principled methodologies to build ontologies, we also paid attention to the issues identified by each methodology and its proposed solutions. In particular, we tracked the influences behind the methodologies by identifying (1) if a methodology has included parts from other methodologies within its own approach, and (2) which methodologies have been analyzed to identify open issues. Thus, the following criteria were also extracted for further analysis:

11. **Study of Existing Methodologies**: this criterion identifies which existing methodologies have been studied or compared with to define the issues to be solved;
12. **Use of Existing Methodologies**: shows if the methodology incorporates parts of existing methodologies into their own approach.

Upon completion of Phase 2, a list of selected studies was available for analysis. The relevant information was extracted and copied onto a Word document, using a template containing the coding categories. Each methodology was uniquely identified by either name (if available) or author(s).

4.3.7 Synthesis strategy

The synthesis of the selected studies contains both qualitative and quantitative results. The main approach used in this Systematic Review is the qualitative synthesis, which includes primarily narrative descriptions of the researcher's interpretation about the content extracted from the selected publications. The descriptions address the categories of the coding systems (see Section 4.3.6) and how the study answers the review and the appraisal questions. The quantitative synthesis includes a frequency of each time the coding categories are described in the studies, which refers to "how many times a given characteristic or motivator is identified in different papers, not how important it may be" (Beecham *et al.*, 2008, p.863).

The coding system proposed does not have a pre-defined scale of codes because the content related to each category are expected to be identified and clustered as they are discovered. Therefore, each coding category receives a dichotomous value (i.e., 0= no and 1= yes) to indicate whether the study provides content for a specific category, that is, their occurrences. As multiple studies can describe the same methodology, we will group the studies by methodology. In this case, the synthesis will reflect the view of the methodologies, rather than the individual studies, with regard to their contribution to the process of building ontologies for IS.

The synthesis also considers:

- the different approaches adopted in each category
- a matrix of methodologies that reuse parts from other methodologies
- a summary of the analysis
- potential categorization of the main focus of the methodologies

4.4 Identification of studies

The query results reported in Figure 21 reflect the search performed on 04/04/2009 with a total of 2025 publications retrieved from six major bibliographic databases. The amount of publications reported represents the total of publications stored in the EndNote libraries, rather than the publications retrieved. During the process of importing the extracted files, EndNote identified duplicated publications that were left out of the libraries. For instance, SpringerLink resulted in nine and Proquest resulted in eight duplicated publications.

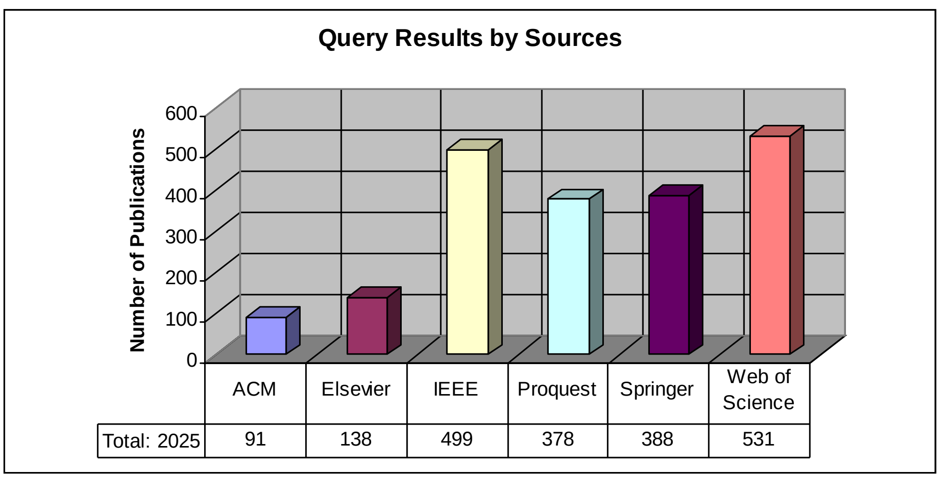

Figure 21: Query results by sources of bibliographic databases

4.5 Selection of primary studies

The screening process for Phase 1 (see Section 4.3.4) is the process of reading all titles and abstracts, and applying the inclusion and exclusion criteria to evaluate if the publications could potentially be relevant to the review and should move on to the next phase of selection.

For each citation selected to go to Phase 2 (see Section 4.3.4), its respective paper was retrieved from the electronic database for further investigation. In cases where the paper was not available, a manual search was performed using alternative electronic sources, such as Google Scholar or Citeseer. If the paper was still missing, an inter-library loan would be requested through the Penn State Library. Several papers ended up being acquired through this process, and two papers could not be obtained.

In Phase 2, the inclusion and exclusion criteria were applied to the body of the papers selected in Phase 1. If the papers pass the selection criteria, they were included into a new spreadsheet of the final candidates.

Not all methodologies shown in Cardoso's (2007) survey have been included in the final list of methodologies. According to Petticrew & Roberts (2006) "no search is complete if it does not also include searching the bibliographies of a selection of key traditional reviews and major discussion papers. This may uncover studies that have not been listed in electronic databases" (p.102). Thus, we also included survey papers from the literature as sources of publications (Breitman *et al.*, 2007; Corcho *et al.*, 2003; Fernandez-Lopez, 1999; Fernandez-Lopez & Gómez-Pérez, 2002; Gómez-Pérez, 1999; Gómez-Pérez *et al.*, 2004; Pinto & Martins, 2004).

After checking the citations to find the primary studies, and applying the inclusion and exclusion criteria to the papers, eight new methodologies were added to the final list of methodologies to review (Table 7). These methodologies are indicated with the source "Survey".

Table 7: List of selected methodologies

#	Year	Methodology	Source	Reference
1.	1990	OBCM/IS	Proquest	(Takagaki, 1990)
2.	1990	CYC	Survey	(Lenat & Guha, 1990)
3.	1995	Grüninger & Fox	Survey	(Grüninger & Fox, 1995; Uschold, 1996; Uschold & Grüninger, 1996)
4.	1995	Kactus	Survey	(Schreiber *et al.*, 1995)
5.	1995	Uschold & King	Survey	(Uschold, 1996; Uschold & Grüninger, 1996; Uschold & King, 1995)
6.	1996	Methontology	Survey	(Fernandez *et al.*, 1997; Gómez-Pérez, 1998; Gómez-Pérez *et al.*, 1996)
7.	1997	SENSUS	Survey	(Swartout *et al.*, 1997)
8.	1999	CAKE	Web of Science/Springer	(Gavrilova *et al.*, 1999)
9.	2000	KIS/OBM	IEEE	(Wang & Xu, 2000)
10.	2001	Chen & Chan	Springer	(Chen & Chan, 2001)
11.	2001	Noy & McGuinness	Survey	(Noy & McGuinness, 2001)
12.	2001	On-to-Knowledge	Proquest\Survey	(Staab *et al.*, 2001)
13.	2002	Bachimont et al.	Springer	(Bachimont *et al.*, 2002)
14.	2002	Heuristic-Based	Web of Science/Elsevier	(Sugumaran & Storey, 2002)
15.	2002	Bowman	Proquest	(Bowman, 2002)
16.	2003	Lexicon-based	Survey	(Breitman & Leite, 2003)
17.	2004	Jin et al.	IEEE	(Jin *et al.*, 2004)
18.	2005	Dilligent	ProQuest	(Vrandecic *et al.*, 2005)
19.	2005	Gavrilova & Laird	Springer	(Gavrilova & Laird, 2005)
20.	2005	Husseman & Vossen	Web of Science/Springer	(Hüsemann & Vossen, 2005)
21.	2006	Brusa et al.	ACM	(Brusa *et al.*, 2006)
22.	2006	Kong et al.	IEEE	(Kong *et al.*, 2006)
23.	2006	DKAP	Proquest/IEEE	(Sarder *et al.*, 2007; Sarder, 2006)
24.	2006	Tun & Tojo	Springer	(Tun & Tojo, 2006)
25.	2007	O4IS	ProQuest	(Vandana, 2007)
26.	2007	Ahmed et al.	Web of Science	(Ahmed *et al.*, 2007)
27.	2008	Jung et al.	IEEE	(Jung *et al.*, 2008)
28.	2008	Sureephong et al.	Springer	(Sureephong *et al.*, 2008)
29.	2008	OE	Springer	(Huang *et al.*, 2008)
30.	2008	ROC	Springer	(Koenderink *et al.*, 2008)

From this point onward, the methodologies are identified by the names shown in Table 7, and are written with the font type Courier New (e.g., `Noy & McGuinness`) to distinguish the name of the methodology from the citation of the paper describing the methodology.

Some of the electronic databases used as source to find publications are known to present results from different sources. For instance, a search performed on the Web of Science can result in publications from IEEE Xplore, ACM Digital Library, SpringerLink. Figure 22 shows the development timeline of the selected methodologies. The timeline refers to the year of the publication presenting the methodology, rather than the year when the methodology was created.

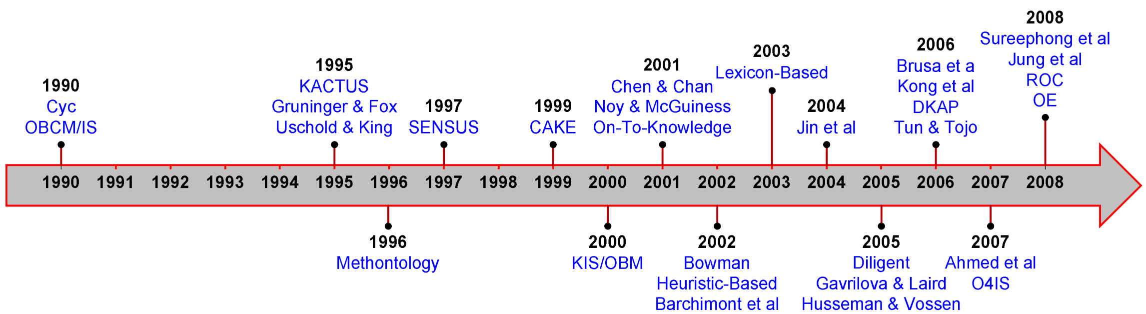

Figure 22: Development timeline of the selected methodologies

Figure 23 shows the amount of publications selected from each database, including the overlaps.

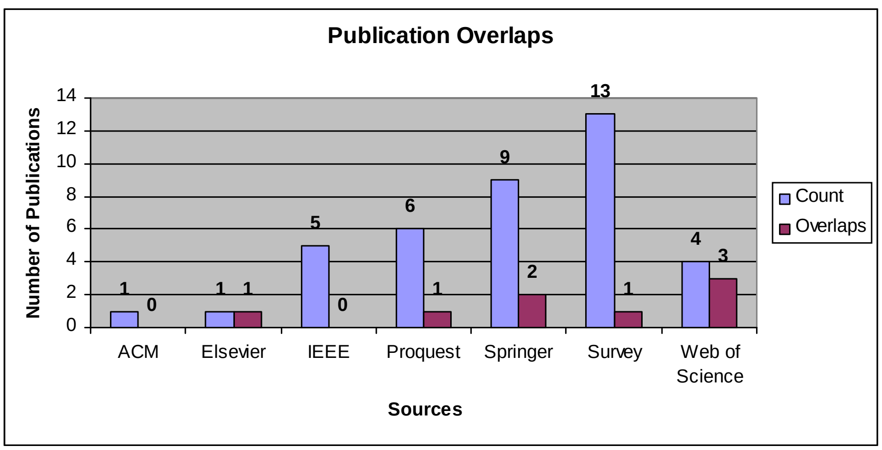

Figure 23: Publication overlaps by source

4.6 Quality assessment

Each study was evaluated according to the appraisal questions, and the result was computed and grouped by methodologies. The possible rates are 0 = *N/A*, 1 = *List*, and 2

= *List and Describe*. The studies were evaluated on their contribution to the systematic review, rather than on the quality of the research. The contribution refers to the expected content towards building an ontology for Information Systems.

The results of the quality assessment are presented in Chapter 6 when we discuss the contribution of the methodologies to the process of building ontologies for IS.

4.7 Data extraction and monitoring

The data extraction from the selected papers was recorded as a form on a Microsoft Word Document. A template was used to organize the extracted content (see Table 8). The template includes the 12 categories described in Section 4.3.6 and other categories that can help "to collect all the information needed to address the review questions and the study quality criteria" (Kitchenham, 2004, p.17).

Table 8: Template of a data extraction form

Methodology Name:	The name or acronym of the methodology
Authors:	The name of the author(s) of the paper
Title:	The title of the paper
Year:	The year of the publication
Source:	The name of the electronic database where the paper was retrieved from
Knowledge Acquisition:	Methods for acquiring domain knowledge
Identify Concepts:	Methods to identify and capture concepts
Identify Relationships:	Methods to identify and capture the relationships between the concepts
Identify Tasks:	Methods to identify and capture the procedural knowledge of a domain
Identify Temporal Relations:	Methods to identify and capture the temporal relations between tasks
Identify Axioms:	Methods to identify and capture the constraints of a domain
Ontology Levels:	Description of ontology levels used to represent different abstraction views of the domain
Mapping Levels:	Description of how to map and link the different ontology levels
Steps:	Description of the methodological steps proposed to build ontologies
Examples:	Demonstration of the use of the methodology steps through examples
Study Existing Methodologies:	List of existing methodologies used as reference to identify issues
Use Existing Methodologies:	List of existing methodologies and the parts reused within the proposed methodology
Other:	Addition content that can provide important information about the process of building ontologies

4.8 Limitations

The review process recognized limitations that may influence the outcomes of the research:

- Despite suggestions to include additional researchers in the process of screening and other stages of the review (Cooper & Hedges, 2009; Petticrew & Roberts, 2006), this review is conducted as part of a Ph.D. research, and reflects the view of one person.
- The search could potentially find more methodologies if other electronic databases are included into the search strategy, as well as alternative queries. However, the selected databases can be considered a significant representation of publications in the areas of Computer Science, Information Science and Information Systems. In this case, the search strategy aimed to achieve a complete search of the proposed databases, instead of reaching comprehensiveness or saturation (Petticrew & Roberts, 2006).
- Some relevant primary studies may be overlooked because the words used in titles and abstracts do not clearly describe the content of the paper. The initial screening process relied solely on the content of titles and abstracts, and only the publications passing the inclusion and exclusion criteria would be further investigated.
- The analysis of the methodologies relied solely on the primary studies found through the searching process and in some cases through citation search, and may not reflect the most up-to-date content of the methodology.
- As discussed in Section 4.3.5, the limitations of report size imposed by most of publication venues could be a factor threatening the credibility of a primary study, as the same research could present various degrees of in-depth discussion depending on the type and size of the publication used to report the research. Figure 24 shows the different types of publications of the selected primary studies.

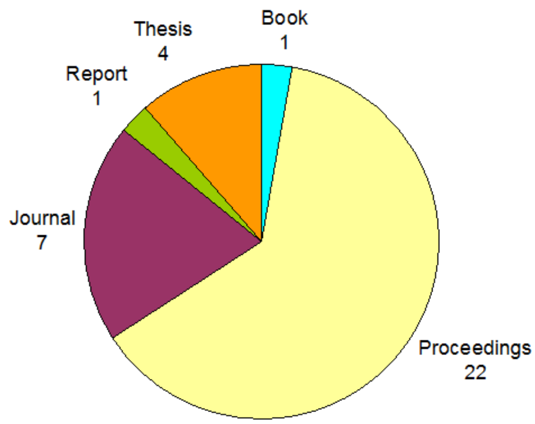

Figure 24: Selected publications by type

4.9 Summary

This chapter presented the process of planning and conducting a Systematic Review as suggested by Kitchenham (2004). Among the most important steps are the clear definition of review questions and the detailed description of the review protocol. A pre-defined review protocol guided the researcher throughout the review process and should reduce the researcher bias to produce a more rigorous and reliable research (Torgerson, 2003). The chapter also presented the list of the primary studies selected (grouped by methodologies), which is synthesized in Chapter 5.

Chapter 5.

SYNTHESIS OF THE METHODOLOGIES SELECTED FOR REVIEW

This chapter presents a synthesis of the 30 methodologies selected for review (see Chapter 4). The methodologies are discussed throughout the chapter with respect to the twelve categories that we developed to investigate methodological approaches to capture domain knowledge, to identify the main components of an ontology, and to address the issues identified in the process of building ontologies for IS design (see Chapter 3).

The overall occurrence of each category by methodology is presented in Figure 25. Each category may include descriptions that range from just a list of approaches to a more detailed description of an approach. So, the count of occurrences is not to be considered a measure of quality.

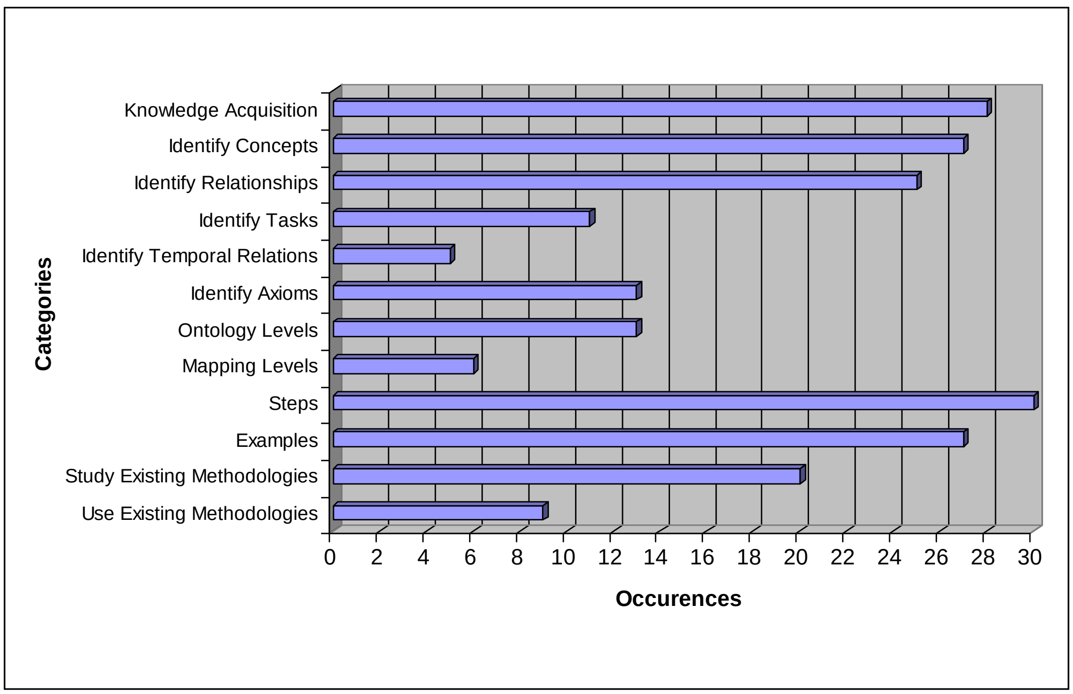

Figure 25: Occurrences per categories

An initial observation of the chart reveals that some methodologies do not cover all the categories. In fact, only the O4IS methodology was represented in all categories. With respect to the issues identified in Chapter 3 (i.e., knowledge acquisition, procedural knowledge, metamodels and temporal relations), only the “Knowledge Acquisition” is included in almost all methodologies. The other categories show a low number of occurrences, which may be an indication that the methodologies have overlooked these issues. Although, we recognize that most methodologies do not claim to be a solution for building ontologies for Information Systems per se.

For the purpose of building such ontologies, most of the methodologies analyzed do not provide appropriate support, especially with regards to procedural knowledge, temporal relations, and ontology levels. Table 9 shows, by methodology, the occurrences of each one of the twelve categories described in Chapter 4.

Table 9: Summary of occurrences

Methodologies	Categories											
	1	2	3	4	5	6	7	8	9	10	11	12
Ahmed et al.	1	1	1	1					1	1	1	
Bachimont et al.	1	1	1			1	1	1	1	1	1	
Bowman	1	1	1	1		1			1	1		
Brusa et al.	1	1	1			1	1		1	1	1	1
CAKE	1	1	1	1	1				1		1	
Chen & Chan	1	1		1			1		1	1		
Cyc	1				1	1	1	1	1			
Dilligent	1	1	1						1	1	1	1
DKAP	1	1	1	1			1		1	1	1	
Gavrilova & Laird	1	1	1						1	1	1	
Grüninger & Fox	1	1	1	1	1	1			1	1		
Heuristic-Based	1	1	1	1	1	1			1	1	1	
Huseman & Vossen	1	1	1	1					1	1	1	
Jin et al.	1	1	1			1			1	1	1	
Jung et al.	1	1	1						1	1		
KACTUS							1	1	1	1		
KIS/OBM	1	1							1		1	
Kong et al.	1	1							1	1		
Lexicon-Based	1	1	1			1	1		1	1	1	1
Methontology	1	1	1			1	1	1	1	1	1	1
Noy & McGuinness	1	1	1			1			1	1		1
O4IS	1	1	1	1	1	1	1	1	1	1	1	1
OBCM/IS	1	1	1	1		1	1		1	1		
OE	1	1	1						1	1	1	1
On-to-Knowledge	1	1	1			1			1	1	1	
ROC	1	1	1						1	1	1	1
Sensus	1	1	1				1	1	1	1		
Sureephong et al.	1	1	1	1			1		1	1	1	1
Tun & Tojo			1						1	1	1	
Uschold and King	1	1	1				1		1	1	1	
Occurrences	**28**	**27**	**25**	**11**	**5**	**13**	**13**	**6**	**30**	**27**	**20**	**9**

5.1 Knowledge acquisition

Previous surveys on the ontology development process (Corcho *et al.*, 2003; Gómez-Pérez *et al.*, 2004) have used knowledge acquisition as an important feature for comparison between methodologies. Knowledge acquisition is one of the main activities in the process of building ontologies. A common approach is a knowledge engineer working with a domain expert to capture relevant knowledge to be represented in an ontology. Nevertheless, this is not the only approach to capture knowledge and certainly not the only activity involved in this process.

According to Milton (2007), knowledge can be described and seen in different ways (i.e., conceptual vs. procedural knowledge, and tacit vs. explicit knowledge). Therefore, methodologies proposing knowledge acquisition, should present detailed information about the involved activities.

In terms of the methods to acquire knowledge, we have identified three main approaches:

- Methodologies that adopt existing methods for knowledge elicitation, such as text and document analysis, interviews with users and observations of their work routines, and for knowledge modeling, such as concept maps or other diagrams.
- Methodologies that adopt other methodologies for knowledge acquisition. For instance, the `Uschold & King` methodology selected the BSDM method developed by IBM as their main method for ontology capture, the `Sureephong et al.` methodology suggested the use of the ORSD (Ontology Requirement Support Document), and the `OE` methodology uses the 5W1H (What, Who, When, Where, Why, How) method to generate domain questions.
- Methodologies that develop and propose new methods targeting the activities of knowledge acquisition. For instance, the `O4IS` methodology introduces the Unified Semantic Procedural Pragmatic (USP^2) Design for domain conceptualization, which includes the Semantic Analysis Representations

(SAR), a mechanism for identifying structural, functional, temporal, prescriptive and Deontic relationships.

Breitman & Leite (2003) warn that "available methods for ontology construction […] concentrate in the modeling aspects and are either vague or lacking on how concepts and relationships are to be elicited" (p.4). Vandana (2007) also points out that the function requirements (i.e., actions and their constraints) of Information Systems are overlooked by ontology design. These warnings still hold for some of the methodologies being reviewed in this research. Thus, the methodologies can also be differentiated with regard to their emphasis on how to acquire knowledge (i.e., the process) or what knowledge to acquire (i.e., the content).

Knowledge is usually extracted from domain experts and described as textual narratives. Alternative notations such as concept/topic maps can be used to model the knowledge as well. Scenarios and competency questions are the most used approaches to elicit knowledge from experts. The first methodology proposing the use of motivating scenarios and competency questions was `Grüninger & Fox`'s, nevertheless, several other methodologies have included this approach. Scenarios are descriptive story problems about the domain (Grüninger & Fox, 1995), and competency questions are questions about the domain that the ontology should be able to answer with its embedded knowledge (Noy & McGuinness, 2001). The `Heuristic-based`, `Husseman & Vossen`, `O4IS`, and `Jung et al.` methodologies adopted a more formalized view, such as use case scenarios.

For most of the methodologies, knowledge acquisition is not a single activity. Usually, the knowledge acquisition step is preceded by an initial activity, often called specification that defines the domain of study, the purpose and scope of the ontology, the sources of knowledge. In some cases, the specification also includes defining the type of ontology to be created and its level of formality, the methods for knowledge acquisition, and the representation and implementation languages to be used. The results of a specification phase can greatly influence and direct the selection of methods for performing the knowledge acquisition.

According to Fernandez *et al*. (1997), the main reason for conducting knowledge acquisition activities in the process of building ontologies is to be able to create a glossary of relevant terms about the domain, and the relationships between terms. The methodological approaches for identifying these terms/concepts are discussed in the next session.

5.2 Concepts and relationships

Identifying concepts and relationships is an interactive process, where concepts can lead to finding new relationships between concepts, and relationships can lead to finding new concepts. Hence, this section will combine the discussion of these two categories to show this interplay.

A common approach for identifying concepts (also called terms or classes), about a domain is to have users and domain experts identify and enumerate things of interest. However, this process puts pressure on users to actually define what is relevant about the domain. Knowledge acquisition approaches, such as scenarios and competency questions (Grüninger & Fox, 1995; Uschold & King, 1995), can provide the users with a guideline for identifying concepts and their relationships. Moreover, the text generated by scenarios and competency questions is the source of information for finding concepts and their relationships. The relevant concepts can be identified by domain experts, knowledge engineer, or some automated application.

Alternatively, knowledge engineers can be the ones responsible for identifying the relevant domain concepts by learning from users and domain experts through brainstorming sessions, interviews or observations. Documents and other natural language sources can also be used to identify concepts, but in this case, knowledge engineers may choose to apply text analysis mechanisms for automatic extraction of concepts. The `Heuristic-based` methodology, for example, suggests the identification of the basic terms by calculating their frequency.

`Noy & McGuinness` suggest that "concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain" (Noy & McGuinness, 2001, p.4). This view is observed in three methodologies. First, the `Lexicon-based` uses lexicon terms represented as object, subject, verbs, or state. Second, the `Methontology` proposes the construction of a glossary of terms that include concepts, instances, verbs and properties. Finally, the O4IS adopts the verb phrase ontology proposed by Storey & Purao (2004), which is an approach "for understanding the semantics of relationship verb phrases by mapping verb phrases to various categories that capture different interpretations" (Vandana, 2007, p.120).

The use of large lexicons has also been proposed to identify new concepts and relationships. The SENSUS methodology uses what they call a broad coverage ontology and the Kong et al. methodology uses the WordNet. Both methodologies suggest linking the initial terms identified by the experts with terms in the lexicon base, and then extracting the concepts related to the initial terms. SENSUS in particular suggests extracting all terms between the start terms and the root of the broad coverage ontology. The Cyc methodology also relies on a large knowledge base.

Uschold & King describe three strategies for defining terms: (1) the top-down approach starts with the definition of few general terms following with their specializations, (2) the bottom-up approach starts with the definition of large number of specific terms following with their generalization, and (3) the middle-out approach starts with the definition of basic concepts following with their generalization and specialization. Uschold & King (1995) argue that the middle-out approach is the most promising of the three. Nonetheless, Noy & McGuinness provide a different view. For them, "none of these methods is inherently better than any of the other. The approach to take depends strongly on the personal view of the domain" (2001, p.7). This view is recommended by several methodologies which suggest these strategies as a way to identify concepts and their hierarchical relationships.

Maedche (2002), cited in (Breitman & Leite, 2003), defines ontology with two types of relationships: taxonomic (i.e., hierarchical) and non-taxonomic. The major emphasis is with respect to hierarchical relationships. In this case, the most common relations used are "SubClassOf", "IsA", and "IsKindOf". The remaining methodologies identify a variety of non-taxonomic relations, such as temporal relations, part-whole relations, and others. The level of detail on how the relationships are described varies from specific descriptions of the type of relations to be used to just an indication of the need for relations, without any specifics on how to define them.

Methodologies committed to upper-level ontologies present an additional framework to help users to identify concepts and relationships, as the constructs of the upper ontology provide a view (i.e., metamodel) on how to see the domain. This is the case of the `OBCM/IS` methodology that adopts Bunge's ontology "as a basis for an IS conceptual model" (Takagaki, 1990, p.45). Bunge's ontology provides a viewpoint of the domain under investigation without commitment to any technical approach (Takagaki, 1990), and provides a set of fundamental constructs to describe reality and to represent a system.

The result of identifying and capturing concepts and their relationships is usually the creation of a partial ontology of the domain that will likely pass for several refinements as users, experts and engineers learn more about the domain. Different methodologies refer to this ontology as initial ontology, baseline ontology, core ontology, or proto-ontology.

5.3 Tasks

An important, and usually neglected, domain knowledge to be included in an ontology for Information Systems is the procedural knowledge, which represents the prescriptive behavior in a domain (Vandana, 2007). This is a knowledge about "processes, tasks and activities", especially about how to perform tasks and related sub-tasks (Milton, 2007, p.4). However, 19 out of 30 methodologies did not provide support to identify and represent this kind of knowledge.

From the methodologies providing such support, some describe the constructs used to represent the procedural knowledge, and others describe the process for identifying the procedural knowledge. For instance, the `OBCM/IS` and the `Cyc` methodologies describe constructs (e.g., event, process, state, etc.) that can be used to represent procedural related knowledge, while the `Grüninger & Fox` methodology suggest the use of competency questions related to planning and scheduling, which refers to "what sequence of activities must be completed to achieve some goal?" (Grüninger & Fox, 1995, p.4). In addition, this knowledge can be extracted from scenarios (or use case scenarios) as suggested by the `Heuristc-based` and the `O4IS` methodologies.

Methods to identify procedural knowledge are suggested as part of the methodological steps to build ontologies. `O4IS` proposes a Procedural Concept View to describe “the knowledge (procedural knowledge) about the emotional states, intentions, plans and rules” of the domain (Vandana, 2007, p.118), `CAKE` proposes the use of a stratification process, which includes the HOWTO-Knowledge (i.e., functional analysis) and the WHY-knowledge (i.e., causal analysis), and `Bowman` suggests the use of “task reduction” to represent the concept of problem decomposition to solve problems. The decomposition occurs by partitioning the high-level problem into smaller sub-problems (Bowman, 2002). The `O4IS` methodology presents the most comprehensive approach for the analysis and representation of procedural knowledge with the SAR:Functional Relationship, which is based on the Verb-Phrase Ontology (Storey & Purao, 2004) to identify actions and its dependences. In addition, `O4IS` uses the ECA (Event-Condition-Action) Rule combined with the 5Ws (i.e., who, what, where, when, why) to identify and to represent procedural knowledge.

The use of multi-level ontology is also used to frame the domain under investigation. The `Chen & Chan` methodology proposes a Process Ontology as part of their Upper Model ontology. Similarly, `Sureephong et al.` proposes a Task Ontology “that specifies terminology associated with the type of tasks and describes the problem solving structure of all existing tasks” (Sureephong *et al.*, 2008, p.336). In addition, the `Grüninger & Fox` methodology suggests an Activity Ontology to define actions that are based on the discrete situation calculus. Situations are presented as trees, which can show “all possible ways in which the events in the world can unfold” (Grüninger & Fox, 1995, p.5). However, these methodologies do not describe the process of identifying the procedural knowledge and mapping it into the constructs of the upper level ontologies.

5.4 Temporal relations

Temporal relations are directly related to procedural knowledge. They describe the temporal logic constraints between tasks. Despite this integration, and the fact that 11 methodologies have provided some support to the process of identifying tasks, only five methodologies provided support to the process of identifying temporal relations.

The `O4IS` methodology bases their SAR:Temporal Relationships approach on the linear temporal logic theory, which describes the relation between two events (e.g., event A starts before event B). It uses the primitives *follow*/*precede* and requires to represent dependencies between two actions. The `Grüninger & Fox` methodology identifies temporality through a set of informal competency questions called Temporal Projection, which is one of the set of questions to create the Activity Ontology (i.e., actions performed). For instance, "given a set of actions that occur at different points in the future, what are the properties of resources and activities at arbitrary points in time?" (Grüninger & Fox, 1995, p.3). The `CAKE` methodology suggests the strata WHEN-knowledge for Temporal Analysis of the domain. That would include "Schedules, Time Constraints, etc." (Gavrilova *et al.*, 1999, p.753).

Temporal constraint is also proposed by the `Heuristic-based` methodology as part of the heuristic-3.2. This heuristic states that "one term/relationship must occur before another" (Sugumaran & Storey, 2002, p.259). The methodology also includes heuristics for mutual inclusive constraint (i.e., heuristic-3.3), where "one term/relationship requires another for its existence" (p.259); and for mutual exclusive constraint (i.e., heuristic-3.4), where "one term/relationship cannot occur at the same time as another" (p.260). Finally, the `Cyc` methodology does not present a specific approach to identify temporal relations, but it describes constructs that can be used to represent these relations.

5.5 Axioms

The literature shows ontologies with different definitions and with different number of components, such as a 4-tuple ontology (Stevens *et al.*, 2000) and a 5-tuple ontology (Maedche, 2002), but they all have "axioms" as a common and important component of the ontology. `Grüninger & Fox` argue that "simply proposing a set of objects alone, or proposing a set of ground terms in first-order logic, does not constitute an ontology. Axioms must be provided to define the semantics, or meaning, of these terms" (Grüninger & Fox, 1995, p.7). In addition, they consider that "the process of defining axioms is perhaps the most difficult aspect of defining ontologies" (p.7). Falbo *et al*. (2002, 1998) also warn about the fact that an ontology is not just a conceptualization of concepts and hierarchies, but a "fully axiomatized theory about the domain" (Falbo *et al.*, 2002, p.352).

While axioms are commonly accepted as a core component of an ontology and defined as constraints, rules or restrictions of a domain, the distinction between possible types of axioms has proved to be not so clear, especially with regard to the process of identification and representation of axioms.

The approaches around axioms can be separated into three categories:

- methodologies describing ontological constructs, predicates or other structures to represent the axioms
- methodologies suggesting methods/processes to identify axioms
- methodologies suggesting logic representations to define axioms (e.g., first-order logic)

Some methodologies list structures that can be used to represent axioms. Both the `Noy & McGuinness` and the `Jin et al.` methodologies propose facets to allow restrictions on slots (i.e., properties). Facets can be used to describe "the value type, allowed values, the number of the values (cardinality), and other features of the values the slot can take" (Noy & McGuinness, 2001, p.9). Also, the `Methontology` suggests the creation of tables of axioms, formulas, and rules to represent restrictions and controls in the domain.

In `Bachimont et al.`, a Referential Ontology provides the structure to represent axioms. The Referential Ontology is the result of a transition from the Differential Ontology created in the first step of the methodology, where new concepts in the Referential Ontology represent extensions of the concepts in the Differential Ontology. Their approach proposes to augment the ontology by adding "logic axioms in relation to relational algebra, part-whole reasoning, composition of relations, exhaustive partitions, etc." (Bachimont *et al.*, 2002, p.119) citing (Staab & Maedche, 2000).

Other methodologies describe specific constructs to represent axioms. For instance, the `OBCM/IS` describes a construct called "Law statements" to define constraints and rules. The `Heuristic-based` proposes the basic constraints (i.e., pre-requisite, temporal, mutual inclusive and mutual exclusive) and the higher-level constraints (i.e., domain constraints and dependencies) to capture business rules. The

Grüninger & Fox methodology uses predicates, such as Poss(a, s), Do(a, s, s'), do(a, s), holds(f, s), actual(s), and occurs(a,s) to describe the relation of an action "a" in a situation "s". These predicates can be use to represent axioms in the Activity Ontology.

In terms of the process to identify axioms, the Grüninger & Fox methodology suggests extracting the constraints from the competency questions and then reusing the competency questions to evaluate the axioms created. They consider that "once informal competency questions have been posed for the proposed new or extended ontology, the terminology of the ontology must then be specified using first-order logic" (Grüninger & Fox, 1995, p.5). For the Lexicon-Based methodology, a list of axioms is generated from the analysis of the connotation of a term (also called behavioral response) in case of an identification of possible disjoint relationships of lexicon terms (i.e., subject, object, verb and state). Brusa et al. consider axioms to be restrictions of classes in the hierarchies of the Concepts Classifier Tree. They argue that "competency questions allow defining a hierarchy so that an answer to a question may also reply to others with a model general scope by means of composition and decomposition process" (Brusa *et al.*, 2006, p.9).

In the O4IS, axioms are defined in the step 5 of the Semantic Concept View. The step is illustrated with constraints of a car: "For modeling the axiom that a sedan can seat only five people. We define a restriction on the seating capacity property of a sedan. We set the 'has Value' option to 5. We also restrict the steering control to a two wheel drive" (Vandana, 2007, p.154). In addition, Vandana advocates the use of Deontic logic, which can be "used to define morality, norms and obligations like constraints that ought to be true" (p.49). The O4IS relies on the ECA (Event-Condition-Action) Rule Ontology to model prescriptive rules.

Axioms have been used to describe different types of restrictions, such as logical axioms (e.g., logical expressions), rules (e.g., if-then condition), and relationship constraints (e.g., cardinality or quantifier restrictions). They can be represented with a selection of logical languages, such as IDEF5 Elaboration Language, First-Order Logic, Second-Order Logic, Description Logic, F-Logic, Higher-order logic, OCL, and CycL.

Even though axioms are a core component of an ontology and a fundamental feature to represent expressive ontologies, our analysis of the methodologies revealed that only 13 methodologies (see Table 9) presented some kind of support to the identification and representation of axioms. Therefore, we consider axioms to be an issue in the process of building ontologies for IS that needs to be addressed by the methodologies.

5.6 Ontology levels

The adoption of levels of ontology is pronounced in several methodologies, usually with the intention to differentiate levels of abstraction. There are no fix numbers of possible levels; nevertheless, a common distinction exists between a top level ontology, a lower level ontology (often called domain ontology), and middle level ontologies.

In the `KACTUS` methodology, Schreiber et al. (1995) consider that “an ontology is formulated as a meta-level theory representing a certain viewpoint on a set of possible domain theories” (p.2), which contain “a set of expressions that represents a model of some application domain” (p.2). They perceive “the relation between a domain theory and an ontology as an object-meta relation”. They also propose that a viewpoint can be achieved through “a set of mapping rules that rewrites a domain theory into a form dictated by the ontology” (p.3).

An example of the use of a top level ontology is the `OBCM/IS`. Takagaki’s Ontology-Based Conceptual Model “operationalizes Bunge’s ontological system so that it can be used to describe some ‘slice of reality’ or Universe of Discourse of an IS application” (p.82). The model includes basic modeling constructs, such as objects and model objects that can be implemented as Ontology-Based Information Systems. By adopting Bunge’s ontology, Takagaki is committed to its ontological assumptions (i.e., views), such as “The world is composed of things”, “Things are grouped into system or aggregates of interacting components”, “Every system, except the universe, interacts with other system in certain respects and is isolated from other systems in other respects”, “Every thing changes”, “Everything abides by law” (Takagaki, 1990, p.56) citing (Bunge, 1977, pp.16-17).

Four methodologies proposed the use of multi-level ontology. First, the `O4IS` methodology calls it a Multi-tiered Domain Ontology that is composed of Upper Generic Ontology, Specific Domain Ontology, and Application / Template Ontology. Vandana

(2007) argues that moving from the top level towards the lower levels is to "move from the generic to specific conceptualization of the target domain" (p.105). Second, the Sureephong et al. methodology proposes three types of ontologies (Sureephong *et al.*, 2008, p.336): Generic ontology is "reusable across the domain", Domain ontology is "defined for conceptualizing on the particular domain", and Task ontology "specifies terminology associated with the type of tasks and describes the problem solving structures of all existing tasks". Third, the DKAP methodology, which targets the product and process domain, proposes a similar structure to indicate the levels of a design ontology (Sarder, 2006, p.59): the site-specific ontology ("for a specific industrial site"), the practice ontology ("models of an entire industry"), and the domain ontology ("information about a general domain"). Finally, the Chen & Chan methodology describes a multi-level ontology composed of three levels: Upper model, Middle Model and Domain Specific Model. Figure 26 shows the integration of the levels in Chen & Chan's approach. The roots of the upper model refer to types of ontologies (i.e., class, process and relation), and the concepts in the middle model "consists of common sense [concepts] relevant to the domain of interest" (Chen & Chan, 2001, p.19).

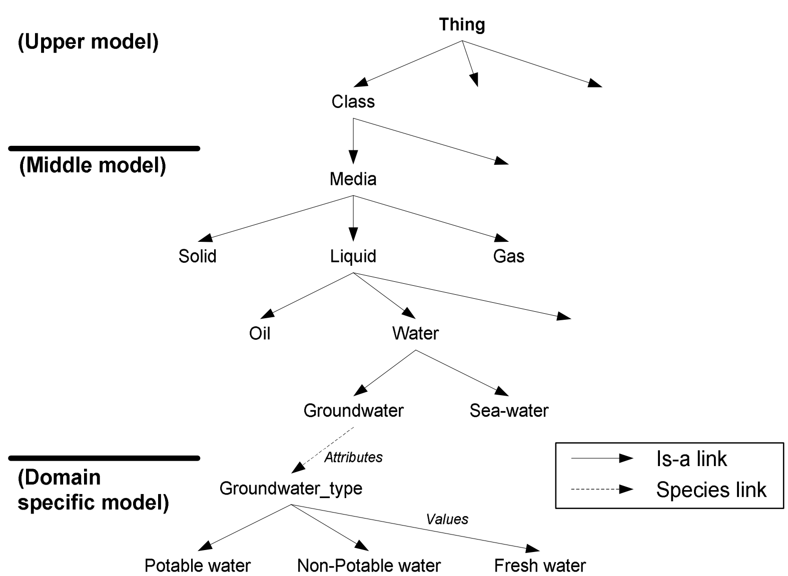

Figure 26: Example of sub-hierarchy of media that illustrates is-a and species links in a category (from Chen & Chan, 2001, p.20)

Sometimes different types of ontology are referred to as ontology levels when in reality they portray the grouping of some ontological features or specific parts of a domain. An example is the `Bachimont et al.` methodology that proposes three levels of semantic descriptions of knowledge primitives: "a linguistic semantic description that provides a human user with an unambiguous understanding of a term; a formal semantic description that provides a human user with a mathematical and formal account of the previous level; a computational description that makes explicit the intended behavior of the computer when handling with this primitive" (Bachimont *et al.*, 2002, p.116). The semantic normalization step generates the Differential Ontology, which is imported into the Referential Ontology in the formalization step. And, the operationalization step converts the Referential Ontology into a Computational Ontology.

Large lexicons are also interpreted as meta-ontology. The `Methontology` suggests the reuse terms from a meta-ontology, such as Ontolingua as inputs for the domain ontology. Likewise, `SENSUS` and `Cyc` link terms from a large ontology into terms of a domain ontology. However, `Cyc` has a top level ontology, called Cyc's Global Ontology, that frames the rest of the ontology. For `Cyc`, the world "is composed of things related to each other" (Lenat & Guha, 1990, p.173), and "these things are grouped into different collections" (p.173), such as Event, Process, Intangible object, etc. The thousands of terms in the `Cyc` ontology are based on these top level categories (Sowa, 2000).

5.7 Mapping between ontology levels

When a methodology adopts different levels of ontologies, the levels above become a grammar (i.e., frame) to be mapped into the levels below, which requires some guidelines for properly identifying what constructs to use. The mapping from domain concepts to grammar constructs are called representation mapping (Wand, 1996). Examining the mappings can identify potential problems with the constructs (Fettke & Loos, 2003; Wand, 1996). Figure 27 illustrates the four problems with mapping.

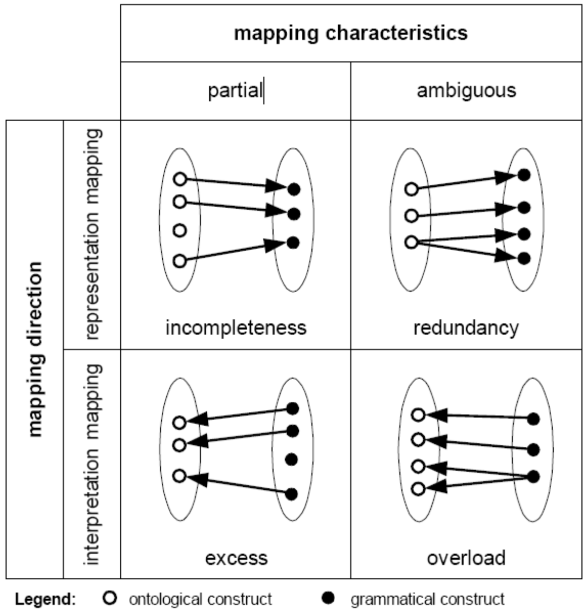

Figure 27: Ontological deficiencies of a grammar (extracted from Fettke & Loos, 2003, p.2948)

- *Ontological incompleteness* refers to the shortage of constructs to represents the domain concepts;
- *Construct overload* occurs when one construct is used to represent more than one domain concept;
- *Construct redundancy* occurs when more than one construct can be used with the same domain concept;
- *Construct excess* means that some constructs will not have a match to a domain concept.

Not all methodologies that presented ontology levels provide a description of the process of mapping between the levels. The most common approach is linking terms from the upper level with terms from the level below.

The KACTUS methodology proposes the use of mapping rules. According to Schreiber *et al*. (1995), the mapping can be operationalized in two ways: "the first one is the mapping of the vocabulary of one ontology onto the vocabulary of the other ontology without a change in the semantics of the expressions. The second one entails a change in the semantics of the ontology" (p.6). An example of the first mapping is of a "*boat* in one ontology […] mapped on *ship* in another ontology if they refer to the same type of objects in the universe of discourse" (p.6). For the second mapping, "an example is the mapping of the concept *variable* in a *mathematical-formula* ontology on the concept *parameter* in a *state* ontology" (pp.6-7).

Similar to the mapping approach, the Methontology suggests the development of an integration document (see Table 10), which is a kind of translation mechanism to link terms from different levels.

Table 10: An example of an integration document (from Fernandez *et al.*, 1997, p.38)

Meta-Ontology	The frame-ontology in Ontolingua	
Term in your conceptualization	**Ontology to be reused**	**Name of the term in the ontology**
Kilometer	Standard-Units in Ontolingua	Kilometer
Centimeter	Standard-Units in Ontolingua	Undefined
Exponent	KIF-Numbers in Ontolingua	Expt

The Bachimont et al. methodology describes the import from and the export to different types of ontologies, which denotes the reuse of ontologies. For instance, the methodology suggests importing the Differential Ontology into the Referential Ontology, and then in another step, it suggests exporting the Referential Ontology into a Computational Ontology. Reuse of ontologies is also demonstrated in the example of a Specific Domain Level Contract Ontology, given in the O4IS methodology, where the mapping occurs through the "extension or restriction of the upper level core contract ontology to a specific domain [ontology]" (Vandana, 2007, p.196).

5.8 Methodological steps

According to Husemann & Vossen (2005), "the ontology design approaches reported in the literature can be put in two major categories, based on whether they adapt methodologies from Knowledge Management or from Software Engineering" (p.50). This categorization was observed among the methodologies investigated by this review as well.

The methodologies based on Knowledge Management can be compared to Buchanan's general model for knowledge modeling (Bowman, 2002) citing (Buchanan *et al.*, 1983), presented in Figure 28. The methodologies based on Software Engineering can be compared to Nunemaker's Software Development Research Process (Nunamaker & Chen, 1990), presented in Figure 29.

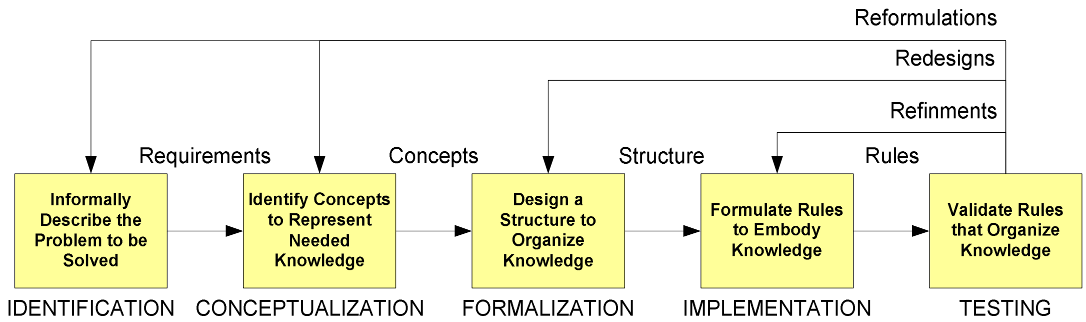

Figure 28: A general model for knowledge modeling (adapted from Bowman, 2002) citing (Buchanan *et al.*, 1983, p.21)

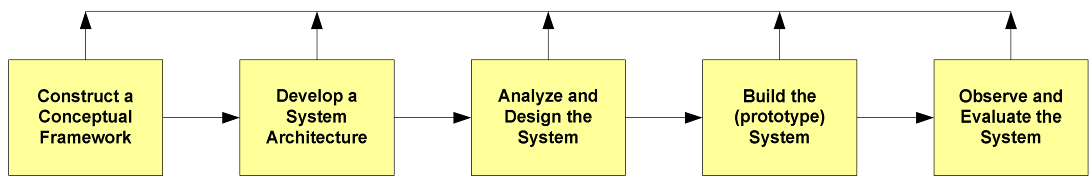

Figure 29: A research process of system development research methodology (adapted from Nunamaker & Chen, 1990, p.636)

Husemann & Vossen (2005) propose a new category. Their novel methodology "is based on the four-phase model of traditional database design consisting of requirements analysis, conceptual design, logical design, and physical design" (p.49). Nevertheless, they state that their approach is a combination of "methods from knowledge engineering (e.g., competency questions) with methods specific to software engineering (e.g., use case diagrams) and to database engineering (e.g., aggregation/generalization diagrams)" (p.61).

Pinto & Martins (2004) present a commonly accepted stages in the process of building ontologies:

- "**Specification**: Identify the purpose and scope of the ontology. Purpose answers the question 'Why is the ontology being built?' and scope answers the question 'What are its intended uses and end users?'";
- "**Conceptualization**: Describe, in a conceptual model, the ontology to be built, so that it meets the specification found in the previous step". In addition,

"the conceptual model of an ontology consists of concepts in the domain and relationships among those concepts. Relationships enhance stronger connections between groups of concepts. These groups of highly connected concepts usually correspond to different modules (subontologies) into which the domain can be decomposed";

- **"Formalization**: Transform the conceptual description into a formal model, that is, the description of the domain found in the previous step is written in a more formal form, although not yet its final form. Concepts are usually defined through axioms that restrict the possible interpretations for the meaning of those concepts. Concepts are usually hierarchically organized through a structuring relation, such as is-a (class-superclass, instance-class) or part-of";
- **"Implementation**: Implement the formalized ontology in a knowledge representation language. For that, one commits to a representation ontology, chooses a representation language and writes the formal model in the representation language using the representation ontology";
- **"Maintenance**: Update and correct the implemented ontology" (Pinto & Martins, 2004, pp.442-443).

With regard to methodological steps, the review observed that most methodologies included steps that resemble the stages presented above. While some of the steps just present a partial description or high-level steps, others include more details, which usually include sub-steps and examples. `Tun & Tojo`, for example, skips the specification stage, and concentrates on the conceptualization and formalization of the ontology.

`SENSUS`, `Lexicon-Based`, `Kong et al.`, and `Tun & Tojo` presented algorithm like steps for the creation of specific parts of the ontology, such as components or constructs. For instance, the `Lexicon-Based` presents six steps and related sub-steps for defining classes, relationships and axioms. Figure 30 shows the step 3. of the `Lexicon-Based` methodology, which is an example of algorithm like steps.

3. Using the list of lexicon terms classified as either object or subject, for each term:

3.1. Add a new concept to the concept list. The concept name is the lexicon term itself. The concept description is the notion of the term.

3.1.1. For each behavioral response,

3.1.1.1. Check the relation list for a relation that expresses it

3.1.1.2. If there is none, add a new relation to the relation list. The relation name is based on the verb of this behavioral response.

3.1.1.2.1. Verify consistency

3.1.1.3. In the concept list, add a new *rel* to the concept in question. The *rel* is formed by the concept in question + relation (defined in 3.1.1.1) + concept it relates to (the concept is the direct/indirect object of the verb in the behavioral response. It is usually a term in the lexicon and appears underlined).

3.1.1.4. Check for negation indicators in the minimal vocabulary that relate the term to other terms. Analyze the pair of terms in order to identify a possible disjoint relationship.

3.1.1.4.1. If true, add the disjoint relationship to the axiom list.

3.2. Verify consistency

Figure 30: Step 3 of the `Lexicon-based` methodology (from Breitman & Leite, 2003, p.6)

Similarly, in the `Tun & Tojo` methodology, the six methodological steps are used to define concepts (i.e., sorts) and their taxonomic structure (Figure 31).

1. Define a set of sorts, *S*, together with *P*(*s*) for each sort $s \in S$ including ICs.

2. Classify the sorts of *S* into the groups of type, quasi-type, role and sub-role.

(a) First, divide *S* into rigid sorts and anti-rigid sorts concerning equation numbers (7) and (8) given in Section 2.3.

(b) Second, divide rigid sorts into types and quasi-types, and also anti-rigid sorts into roles and sub-roles by the classification given in Section 2.4.

3. By our conceptual constraints, check whether the description of each sort satisfies them or not.

4. According to the subsumption constraints, construct sort hierarchies for *S*.

5. Then, check whether each subsumption relationship satisfies equation number (4) or (6) given in Section2.2, or not.

6. If 'No', then go to Step 1 and repeat the steps to restructure the sorts.

Figure 31: `Tun & Tojo` methodological steps (from Tun & Tojo, 2006, p.426)

In terms of the reuse of existing knowledge and ontologies, we agree with Vandana (2007, p.12) who concludes that methodologies advocating knowledge reuse lack comprehensible instructions for such activities. Vandana also states that "most ontology design methodologies available today propose design phases to build ontology from scratch" (p.12), where "no previous versions or knowledge base or data model exists" (p.62).

Another difference in the steps is related to the components of an ontology (e.g., concepts, properties, relations, and axioms). While some methodologies propose steps that take into consideration the identification and representation of all components of the ontology, some just emphasize the identification of few components, particularly concepts and relations. An example is the `Tun & Tojo` methodology, which focuses solely on the definition of concepts and their taxonomic structures. `Tun & Tojo` state that their "method limits ontologies to be sortal" (Tun & Tojo, 2006, p.430), that is "ontologies that organize sorts in subsumption relationships together with ICs [identity conditions]" (Tun & Tojo, 2006, p.428).

An example of taxonomic structures is seen in the `Bowman` methodology. Because of its focus on task reduction, the methodology identifies mainly concepts and hierarchical relationships. Bowman's research proposes a sequence of procedures and guidelines to modeling a problem as three categories (Bowman, 2002):

- <u>Initial Modeling Procedures and Guidelines:</u> identifies the problem and defines the tasks and sub-tasks until obtaining simpler answers to the questions that generate the sub-tasks (see Figure 32-left).
- <u>Ontology specification Procedures and Guidelines:</u> creates a semantic net with objects identified in the tasks (see Figure 32-right).
- <u>Formalization Procedures and Guidelines</u>: creates formal descriptions for the tasks.

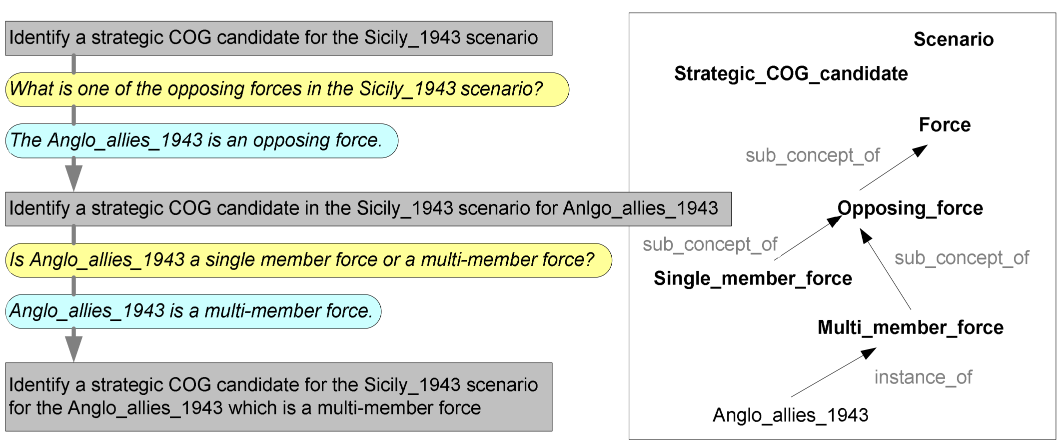

Figure 32: Ontology specification derived from initial domain modeling (from Bowman, 2002, p.70)

Some of the issues regarding the methodological steps have been already discussed in the previous sections of this chapter, such as knowledge acquisition, identification of concepts, relationships, tasks, temporal relations and axioms, and the use of ontological levels.

The last issue to be presented by this analysis of the methodologies is the use of existing approaches or steps as part of a proposed methodology. Nine methodologies reported reusing steps either partially or completely from other methodologies. However, it seems that not all reuse has been reported. The most apparent reuse is the definition of motivating scenarios and competency questions, proposed by `Grüninger & Fox`, to elicit knowledge about the domain. Section 5.11 provides more details about the reuse of methodological approaches.

5.9 The use of examples in methodologies

Most methodologies included examples to illustrate their approach. The use of examples can be distinguished according to the following categories (Figure 33):

- whether or not the methodology presents examples to illustrate the steps;
- if the examples have been implemented and tested in a real setting or if they are a proof-of-concept example;
- if the methodology presents examples after describing all the steps or within the steps;
- if the methodology presents examples to nearly all steps or just some steps;
- if the examples focus on the use of the steps or the ontology created by applying the steps.

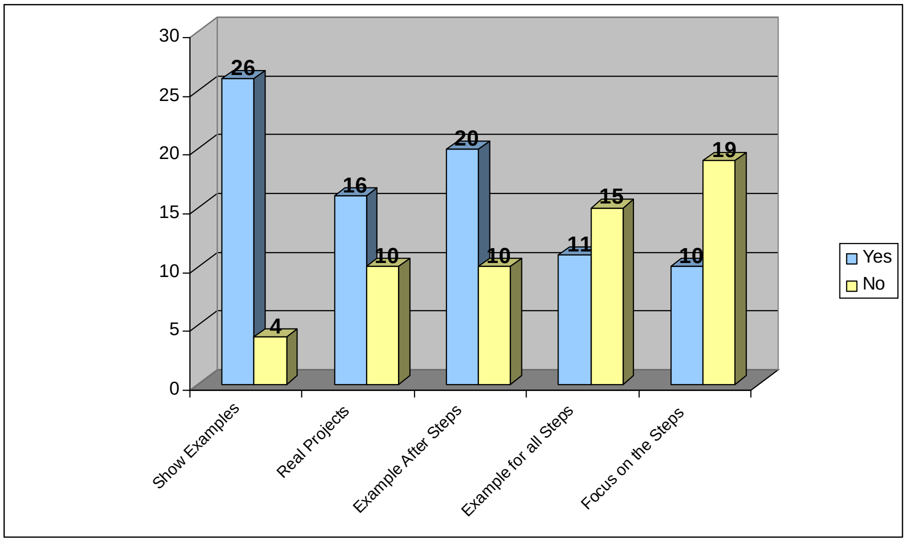

Figure 33: Summary of the use of examples

The most common approaches are to present examples of the use of a methodology in real settings, and to present examples after describing all the steps. Nevertheless, the number of examples and the level of details in the examples vary.

The examples cover a variety of domains. For instance, the `SENSUS` methodology describes the construction of an ontology for a military campaign planning, `Chen & Chan` focus on petroleum waste management, `Noy & McGuinness` present the Wine Ontology, `CAKE` includes some screenshots of a knowledge base for expert system consulting Linux users, and the `Kong et al.` and the `Tun & Tojo` methodologies also use the Wine ontology to describe their work.

5.10 Study of existing methodologies

This category was created with the expectation to find, for each methodology investigated, a list of other methodologies that have been studied to identify open issues, and approaches that can be reused. The results presented in Table 11 show that 20 methodologies have investigated other methodologies; nevertheless, there seems to be a pattern where the same few methodologies are included.

Table 11: Summary of methodologies that study or use other methodologies

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
OBCM/IS	1	Null																													
Cyc	2		Null																												
Gruninger & Fox	3			Null																											
KACTUS	4				Null																										
Uschold and King	5		Study			Null																									
Methontology	6			Study & Use			Null																								
Sensus	7							Null																							
CAKE	8			Study		Study	Study		Null																						
KIS/OBM	9			Study		Study				Null																					
Chen & Chan	10										Null																				
Noy & McGuinness	11			Use								Null																			
On-to-Knowledge	12			Study		Study							Null																		
Bachimont et al	13			Study		Study							Study	Null																	
Heuristic-Based	14												Study		Null																
Bowman	15															Null															
Lexicon-Based	16			Study & Use		Study	Study					Study					Null														
Jin et al	17			Study		Study	Study					Study						Null													
Dilligent	18			Study & Use		Study	Study						Study						Null												
Gavrilova & Laird	19					Study														Null											
Huseman & Vossen	20			Study		Study						Study	Study								Null										
Brusa et al	21			Study & Use		Study & Use	Study & Use					Study & Use										Null									
Kong et al	22																						Null								
DKAP	23											Study												Null							
Tun & Tojo	24						Study																		Null						
O4IS	25			Study & Use		Study & Use	Study & Use					Study & Use														Null					
Ahmed et al	26					Study						Study															Null				
Jung et al	27																											Null			
Sureephong et al	28												Study & Use																Null		
OE	29			Study & Use								Study & Use			Study															Null	
ROC	30						Use					Study																			Null

Legend: Study Use Study & Use Same Methodology (Null)

The discussions about the study of previous methodologies have been presented in different levels of details. For instance:

- the proposed methodology suggests that other approaches exist, but only a few are listed;
- a list of related methodologies to build ontologies is presented, but with little or no description of their approaches and with a summary of what the proposed methodology has to offer or what issues it tackles;
- a list of existing methodologies, including a brief description of their approaches, and a description of issues that the proposed methodology is going to solve;
- a list of existing methodologies that are going to be compared, including a description of their steps and significant features, as well as the approaches that can be reused in the steps of the proposed ontology.

The study of existing methodologies can lead to the adoption of steps or features that are worth to reuse as part of new approaches. Next section discusses the methodologies that have adopted parts/steps from other methodologies.

5.11 Use of existing methodologies

It would be expected from a methodology that is promoting to advance the field of ontology design to build upon previous approaches. However, Table 11 shows that nine methodologies reused steps or features from previous methodologies. Similar to the study of such methodologies, discussed in Section 5.10, reuse is also identified around the same few methodological approaches.

Several methodologies include steps that directly reuse or that resembles the work with competency questions, motivating scenarios, class hierarchy (e.g., top-down, bottom-up), or ontology formality (e.g., informal, semi-informal or formal). `Grüninger & Fox` represents the second most reused approach, followed by the `Noy & McGuinness`, the `Methontology`, the `Uschold & King`, and the `On-to-Knowledge` methodologies.

Table 12 presents a description of which methodology is reusing parts/steps from other methodologies, and for each methodology, which parts are being reused. Similarly to the mapping of the studies of existing methodologies, the mapping of reuse revealed the reuse of other methodologies that are not part of the pool of the selected studies for this review, such as `CommonKADS`, `OntoClean`, `IDEF5`, and `UPON`.

Table 12: Description of the methodological reuse

Methodology	Description of Reuse
Methontology	- The level of formality and the class hierarchy from Uschold & Grüninger.
Noy & McGuinness	- The competency questions of Grüninger & Fox. - The class hierarchy of Uschold & Grüninger.
Lexicon-based	- An approach that corresponds to the competency questions of Grüninger & Fox (p.6).
Diligent	- The competency questions of Grüninger & Fox.
Brusa et al.	- Has phases based on Methontology and tasks based on Noy & McGuinness and Grüninger & Fox (p.13). - Define the goal and scope of the ontology from Noy & McGuiness and Methodology (p.8). - Domain analysis, a list of important terms, and guidelines to identify classes, relations and attributes from Noy & McGuinness. - Intermediate representations for organizing knowledge domain (p.8), and instances definition from Methontology. - Motivating scenarios and competency questions follows Grüninger & Fox. - A template for scenario description and the middle-out strategy for class hierarchy from Uschold & Grüninger.
O4IS	- Ontology formality, and informal knowledge capture, and scenarios from Uschold & Grüninger. - Domain capture from Uschold and King. - Middle-out strategy from Grüninger & Fox and Uschold & Grüninger. - USP[2] design methodology based on Noy & McGuinness and Grüninger and Fox. (p.102). - Competency questions from Grüninger & Fox. - Evolving life cycle of Methontology (p.104). - Noy & McGuinness guidelines to build ontologies, and to capture classes, relations and attributes.
Sureephong et al.	- All steps from On-to-Knowledge.
OE	- Steps based on Grüninger & Fox and Noy & McGuinness (p.496).
ROC	- Variation of Methontology (p.153).

5.12 Summary

This chapter presented a detailed comparison of the methodologies selected for review. Each of the 12 categories was discussed and the highlights and features of the methodologies were presented. Besides exposing the characteristics of the methodologies with regards to the categories, the chapter has shown that the four issues raised in Chapter 3 are also true for the methodologies investigated, and that identifying and representing axioms have emerged as another significant issue in the process of building ontologies.

In Chapter 6, we further the discussion on the initial issues (i.e., knowledge acquisition, metamodels, procedural knowledge and ontology levels) that motivated the review of the methodologies and were used to frame their analysis, and we propose a Scenario-based approach to build ontologies for IS.

Chapter 6.
DISCUSSION

In this research we focused on identifying and reviewing methodologies to build ontologies for Information Systems. We searched bibliographic databases, and we selected 30 methodologies to review. The research analyzed the methodologies with the intention of uncover valuable guiding principles to build ontologies suitable for IS. As reported in Chapter 5, there are many approaches for building ontologies; however, we see opportunities for improvements.

This chapter presents a discussion of the methodologies with regard to their contribution to the systematic review, and their importance for each research question. This chapter also presents a proof-of-concept experiment that shows how the use of a scenario-based approach can help the process of building ontologies for IS. We conclude the chapter with a comparison between the proposed scenario-approach and the top five methodologies from the quality assessment ranking.

6.1 Quality assessment

Each methodology was evaluated according to the appraisal questions, described in Table 6. The appraisal questions related to the issues described in Chapter 3 are Question 1 (i.e., How well is the process of acquiring domain knowledge described?), Question 4 (i.e., How well is the process of representing procedural knowledge described?), Question 5 (i.e., How well are the task dependencies and temporal relations described?), and Question 7 (i.e., How well is the use of ontology levels described?). Each question is answered with a rate of 0=N/A, 1=List, and 2=List and Describe. We computed the average of the rates to provide a ranking of the methodologies with regards to their overall contribution to the review. When the averages are the same, we organize the methodologies by alphabetical order.

Table 13 presents the results of the appraisal questions for quality assessment. The most significant methodology for building ontologies for Information Systems is the `O4IS`, which covers all categories and issues described in this research.

Table 13: Quality assessment of contribution to the systematic review

Methodologies	Quality Assessment Questions										Average
	1	2	3	4	5	6	7	8	9	10	
O4IS	2	2	2	2	2	2	2	2	2	2	**2.00**
Grüninger & Fox	2	2	1	1	2	2	0	0	2	2	**1.40**
Bowman	2	2	2	2	0	1	0	0	2	2	**1.30**
DKAP	2	2	2	1	0	0	2	0	2	2	**1.30**
Heuristic-Based	2	1	1	1	2	2	0	0	2	2	**1.30**
Methontology	2	2	2	0	0	1	1	1	2	2	**1.30**
Brusa et al	2	2	2	0	0	1	1	0	2	2	**1.20**
Bachimont et al	1	1	2	0	0	1	1	1	2	2	**1.10**
Huseman & Vossen	2	2	2	1	0	0	0	0	2	2	**1.10**
Lexicon-Based	1	2	2	0	0	1	1	0	2	2	**1.10**
Noy & McGuinness	2	2	2	0	0	1	0	0	2	2	**1.10**
OBCM/IS	1	1	1	1	0	1	2	0	2	2	**1.10**
Sensus	1	1	1	0	0	0	2	2	2	2	**1.10**
Uschold and King	2	2	1	0	0	0	2	0	2	2	**1.10**
Ahmed et al	2	1	2	1	0	0	0	0	2	2	**1.00**
Sureephong et al	1	1	1	1	0	0	2	0	2	2	**1.00**
Chen & Chan	1	1	0	1	0	0	2	0	2	2	**0.90**
OE	2	2	1	0	0	0	0	0	2	2	**0.90**
On-to-Knowledge	2	1	1	0	0	1	0	0	2	2	**0.90**
ROC	2	2	1	0	0	0	0	0	2	2	**0.90**
Cyc	1	0	0	0	1	1	2	2	1	0	**0.80**
Jin et al	1	1	1	0	0	1	0	0	2	2	**0.80**
Kong et al	2	2	0	0	0	0	0	0	2	2	**0.80**
CAKE	2	1	1	1	1	0	0	0	1	0	**0.70**
Dilligent	1	1	1	0	0	0	0	0	2	2	**0.70**
Gavrilova & Laird	1	1	1	0	0	0	0	0	2	2	**0.70**
Jung et al	1	1	1	0	0	0	0	0	2	2	**0.70**
KACTUS	0	0	0	0	0	0	2	2	1	2	**0.70**
Tun & Tojo	0	0	2	0	0	0	0	0	2	2	**0.60**
KIS/OBM	1	1	0	0	0	0	0	0	1	0	**0.30**

6.2 Research questions

This research identified four issues that should be taken into consideration when creating ontologies to be used for IS. The issues are used to frame the major research questions and to design a systematic review of methodologies to build ontologies. The review focused on identifying approaches that addressed the issues. Following, we discuss the research questions and the main methodological features that can help answering the questions. The focus of the discussion is the building of ontologies for IS modeling, and the outcome is a list of general approaches.

6.2.1 RQ1: Knowledge acquisition

The first research question is “How can we acquire knowledge about a domain?“. Almost all methodologies presented some support to the acquisition of knowledge from the domain. However, knowledge is discussed at different levels of detail. First, with regard to the source of knowledge, the most suggested methods are interviewing and observing domain experts, and performing document analysis. Second, with regard to identifying relevant knowledge from the domain, scenarios/use cases and competency questions are commonly used. Both features can provide flexibility to go from general to specific contexts and vice-versa. However, we argue that a mechanism to support these transitions and to ensure domain coverage (i.e., breadth and depth) should be offered. Third, with regard to knowledge representation, scenarios and competency questions are usually represented as textual narratives with a subsequent process needed to transfer it into the ontologies. Use case scenarios, on the other hand, provide a structure to represent knowledge that can be automatically translated into the ontology. We argue that the domain knowledge should be represented directly into an ontology without the need for an intermediate step that requires translation or transformation from one format to another.

6.2.2 RQ2: Procedural knowledge

The second research question is “How can we identify and represent procedural knowledge?“. Identifying procedural knowledge involves the need to understand how the entities in the domain interact to each other, which means describing the behavior (i.e., the dynamics) of a system, including its events and tasks. According to Milton (2007), procedural knowledge refers to a sequence of tasks needed to achieve a goal, that is, the description of how to do things. We see two high-level approaches for identifying procedural knowledge. First, we can identify the entities in a domain, and then the tasks and events between them. Second, we can identify the events and tasks in a domain, and then the entities that are participating in the tasks. Later in this chapter, we present a proof-of-concept experiment that begins with the identification of events. In terms of representing procedural knowledge, we should consider the appropriate constructs used to represent knowledge about events. The type of constructs used determines the level of details in which events are described. O4IS uses the ECA (Event-Condition-Action) rule combined with the 5Ws (i.e., who, what, where, when, why) to identify and represent procedural knowledge. We recommend the use of components of a scenario (e.g., agents, goals, events and actions) to represent the events happening in a domain. Like O4IS, we also support the use of clause patterns (e.g., Subject-Verb-Object) to express events.

6.2.3 RQ3: Temporal relations

The third research question is "How can we identify and represent temporal relations?". According to Allen (1983), events are related to each other through time intervals, such as before and after. These types of temporal constraints represent the ordering in which events are performed. Therefore, temporal relations are identified from the interplay of events and their dependencies. Temporal Relations were the least explored issues among the methodologies analyzed as only five out of the 30 methodologies presented some approach to support this issue. We argue that the definition of temporal relations should not be detached from the definition of procedural knowledge because they are complementary to one another. With regard to representing temporal relations, we assume that at least a link between the events must be provided to represent their dependencies. In our proof-of-concept experiment, we added a property, called *predecessor*, to the concept Event, which is similar to the primitive *follow*/*precede* used by the O4IS. However, these approaches are not enough to represent all intervals proposed by Allen (1983).

6.2.4 RQ4: Metamodel

The fourth research question is "How can we use meta-ontology to guide the creation of domain ontologies for IS modeling?". By agreeing on a meta-ontology, we establish an ontological commitment to a particular view of the domain under investigation. A meta-ontology is a high-level model (i.e., a metamodel) that can be used as a frame to help build domain ontologies. We argue that connecting a domain ontology to a metamodel ontology enhances our understanding of the domain represented by the ontology. Several methodologies reported using a multi-level ontology approach.

Typically, the levels refer to a top-level domain independent ontology, a middle-level domain dependent ontology, and a lower level domain dependent ontology.

Using a meta-ontology involves building a metamodel ontology that represents some abstract view of the domain, and finding concepts in the domain that can be mapped into the metamodel ontology. The strategy to find relevant concepts is guided by the need to match the content of the metamodel. For instance, we propose a metamodel ontology defined with the components of a scenario. In this case, we consider that an information system is formed by things that interact with other things in a systematic way. So, the strategy to find concepts is to identify the different scenarios within a domain and the different activities within a scenario. Once the concepts are defined, the process of mapping them into the metamodel is also guided by the specific structure of the metamodel, such as events, actions, and agents.

6.3 Axiomatization

Our analysis revealed that only 13 methodologies (see Table 9) presented some kind of support to identifying and representing axioms. Considering that axiom is one of the core components of an ontology, and a fundamental feature to represent expressive ontologies (Falbo *et al.,* 2002; Grüninger & Fox, 1995), we consider the lack of steps addressing axiomatization to be an issue in the building of ontologies for IS. In Ontology-Driven Information Systems, ontology plays a central role in Information Systems as it includes comprehensive descriptions about a domain (Uschold, 2008). For that purpose, without axioms, ontologies would be considered incomplete, as a lot of domain knowledge would be incorporated directly into application programs (Guarino, 1998).

Steps for identifying domain restrictions are still unclear and for the most part limited to basic relationship constraints, such as cardinality and quantifiers. Although the methodologies provide descriptions about structures to capture axioms, ways to identify constraints, and constructs to represent the constraints, there is still no prominent methodology combining these approaches into a set of course of actions.

Despite the excitement with the future of Ontology-Driven Information Systems, where non specialists (i.e., non-ontologists) should be able to represent their own domain with ontologies (Uschold, 2008), we argue that the issue with axioms poses a significant drawback to accomplish this objective. In particular, we consider that writing axioms with logical expressions requires a specific set of skills that is not common to the every-day person. Research in this direction has already been proposed. For instance, EZPAL is a template to help users write axioms in the PAL-Protégé Axiom Language (Hou *et al.*, 2005), and SBVR-Semantics of Business Vocabulary and Business Rules is a standard that propose the use of Structured English to help users to write business rules (Linehan, 2007).

Although we have identified the issue with axioms and we consider axioms to be an important feature for Ontology-Driven Information Systems, exploring the issue further is beyond the scope of this research.

6.4 Scenario-based ontology

The frequent use of *motivating scenarios* (Grüninger & Fox, 1995) in the initial steps of methodologies to build ontologies (discussed in Chapter 5) motivated us to investigate scenarios for the process of building ontologies. In this section, we propose a metamodel ontology based on scenarios, and we present a proof-of-concept experiment to show the use of scenarios in the process of building ontologies for IS. We are particularly interested in approaches that can enhance the representation of domain knowledge in terms of the issues raised in Chapter 3, especially processes, sequence of events, and temporal relations.

According to Grüninger & Fox (1995), "any proposal for a new ontology or extension to an ontology must describe the motivating scenario, and the set of intended solutions to the problems presented in the scenario. This is essential to provide rationale for the objects [and their relations] in an ontology" (p.2). They also consider that through scenarios "we can understand the motivation for the proposed ontology in terms of its applications" (p.2).

We argue that scenarios would be useful for building ontologies for Information Systems because scenarios are stories about people performing some activities to achieve goals (Rosson & Carroll, 2002). Carroll (2000) advocates that an observer, while performing ethnographic studies, is making sense of the domain as he or she "builds an ontology of the agents, goals, actions, events, obstacles, contingencies, and outcomes" (p.257). Similarly, a systems analyst when modeling an IS also makes sense of the domain, and consequently builds an ontology of it, although, as Guarino (1998) mentions, not often explicitly.

Scenarios have been successfully used in IS design for identifying and capturing domain knowledge (Carroll, 2000; Hertzum, 2003; Jarke *et al.*, 1998; Rosson & Carroll, 2002; Sutcliffe, 1998; Weidenhaupt *et al.*, 1998), and have already been adopted for ontology design (Giboni *et al.*, 2002; Grüninger & Fox, 1995; Lee, 2006). However, we argue that scenarios can be further explored if we consider alternative forms of representation. Alternative forms of representation may include, for example, texts, images, diagrams, and videos (Go & Carroll, 2004; Weidenhaupt *et al.*, 1998).

In this research we promote the use of scenarios not only as mechanisms for eliciting knowledge for building ontologies, but also as ontological constructs being part of the ontology itself. A thorough investigation of the components of a scenario revealed a prospective structure for using it as a metamodel ontology. Instead of taking the traditional approach of textual narratives (Alexander & Maiden, 2004; Carroll, 2000; Rosson & Carroll, 2001; Weidenhaupt *et al.*, 1998), the research proposes to use scenarios in the form of computational ontologies, where the components of a scenario become the constructs of a metamodel ontology.

Considering the example of the animal shelter (i.e., PAWS), discussed in Chapter 3, a metamodel ontology based on scenarios would include a construct *Agent* mapping to the domain concept *Cat*, and a construct *Resource* mapping to the domain concept *Cage*. This metamodel ontology will be called *Scenario Ontology (SO)*.

As shown in Figure 34, we envision a metamodel ontology that accounts for the description of scenarios, their goals and their specific events, which would support a description of the procedural knowledge embedded in the domain. The metamodel ontology also provides a frame to view the domain and a guide to create domain ontologies. The dashed lines in Figure 34 represent the mapping between the levels, which can be achieved by adding the domain concepts under the taxonomic structure of the metamodel concepts. This way, we can distinguish, for example, that an *Adopter* is a type of *Agent*, and *Cage* is a type of *Resource*.

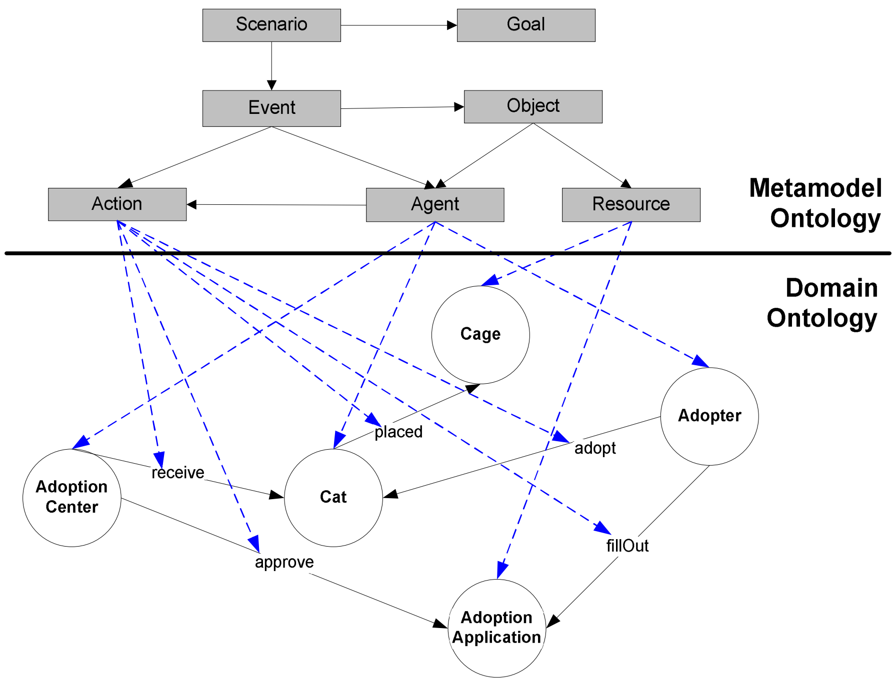

Figure 34: Envisioned mapping between the metamodel ontology and the domain ontology

Our view and proposed use of scenarios is based on the work with Scenarios-Based Design (Carroll, 2000, 1995; Rosson & Carroll, 2002) and on the work of the CREWS (Cooperative Requirements Engineering With Scenarios) project with scenarios-based requirements (Achour, 1998; Rolland & Achour, 1998; Rolland & Prakash, 2000; Rolland *et al.*, 1998; Sutcliffe, 1998). In the next section, we present the structure of the metamodel ontology.

6.4.1 Constructs of the metamodel ontology

A number of structures for scenarios can be found in the literature (Achour, 1998; Alexander & Maiden, 2004; Leite *et al.*, 2000; Rolland *et al.*, 1998; Rosson & Carroll, 2001; Sutcliffe, 1998). The description of a scenario typically includes actors/agents, their goals and objectives within the scenario, and a sequence of actions and events to achieve the goals (Go & Carroll, 2004; Rosson & Carroll, 2002). The metamodel structure (i.e., Scenario Ontology) presented in this section (see Figure 35) is based on the scenario structure presented by Rolland et al. (1998, p.8). However, because this is a proof-of-concept experiment, we do not follow their separation between normal and exceptional scenarios. In addition, we use different labels for some of the components in the scenario structure, we included a property called *predecessor* to help represent the temporality of events, and we add a component to distinguish the actions within an event.

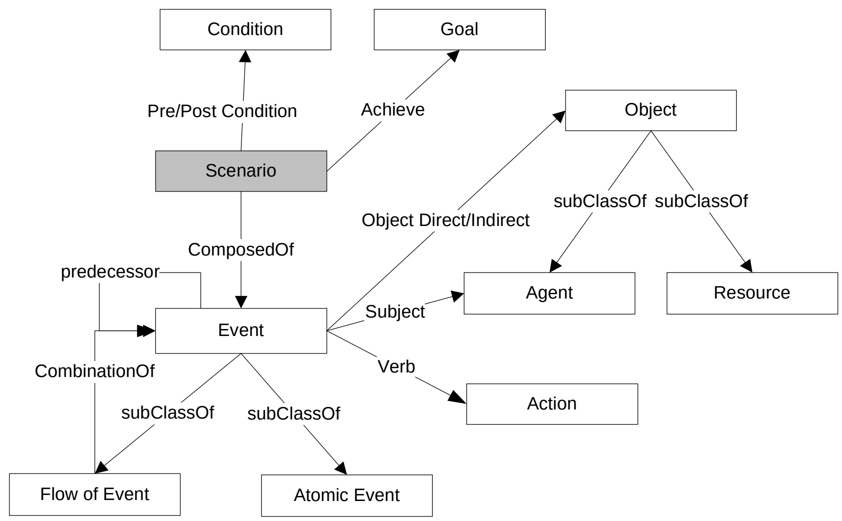

Figure 35: Scenario-based metamodel ontology

A scenario has a goal that is achieved by performing one or more events. A scenario has pre-conditions that initiate the scenario, and a post-condition that is reached upon completion of the scenario. An event can be an atomic event (i.e., a single event) or a flow of events (i.e., an event composed of sub-events). An event can be dependent on the occurrence of other events (i.e., predecessor). The description of an event includes an agent (i.e., subject) that performs an action (i.e., verb) upon an object (i.e., object direct and object indirect), which can be another agent or a resource. We envision the use of subject, verb, object direct, and object indirect to express events of a scenario. This approach is inspired by the use of linguistic clause patterns to express events as a structured English, discussed by Rolland & Achour (1998).

6.4.2 Proof-of-concept experiment

The scenario structure presented in Figure 35 is used to create a proof-of-concept experiment to demonstrate the use of scenarios as a metamodel ontology framing the building of domain ontologies. The experiment is composed of the following parts: (1) build the Scenario Ontology, (2) create the domain ontology (i.e., PAWS Ontology) and import the scenario ontology into it, and (3) represent the different scenarios involved in the domain of the animal shelter by mapping the domain concepts into concepts of the Scenario Ontology. The experiment is implemented with the Protégé Ontology Editor[1], the most used editor to handle the language used to represent the ontology, OWL-Web Ontology Language[2].

The components of the scenario structure are represented as classes of an ontology (see Figure 36).

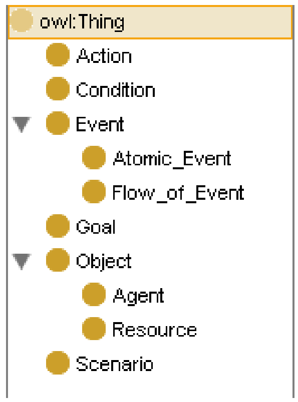

Figure 36: Asserted class hierarchy of the scenario ontology

Atomic Event and Flow of Event are defined as sub classes of the general class Event, and Agent and Resource are defined as sub classes of the general class Object. The relations presented in Figure 35 are defined as object properties (i.e., relations) between classes. In addition, cardinality restrictions are added to the properties of the class Event to enforce that only one agent can perform one action in one event.

Now that the Scenario Ontology is completed, we can create the domain ontology, which will include the domain knowledge about the animal shelter, especially about the life-cycle of a cat at the shelter. The domain ontology is called PAWS Ontology. Next, we import the Scenario Ontology into the PAWS Ontology to provide a frame that will guide the creation of the domain ontology. This way, the Scenario Ontology becomes a metamodel ontology, which reflects a commitment to view the domain as scenarios, where events are performed by agents to achieve goals.

Once the import operation is completed, we can start adding concepts and scenarios from the domain. To help identify the relevant concepts and scenarios, we used

[1] http://protege.stanford.edu
[2] http://www.w3.org/2004/OWL

task analysis to decompose complex events into sub-events, as proposed by Rosson & Carroll (2001). The high-level events represent the different scenarios related to a cat, which can range from the time someone steps in the shelter with a cat to the time an adopter steps out of the shelter with an adopted cat. As for a proof-of-concept experiment, we are just interested in showing the use of scenarios to build ontologies. Therefore, the description of the domain is incomplete and intentionally does not cover alternative scenarios or detailed axiomatization.

The scenario illustrated in this experiment refers to when a cat is brought to the shelter. Figure 37 shows the classes of the domain ontology (i.e., PAWS Ontology) connected to the metamodel ontology (i.e., Scenario Ontology). The concepts from the Scenario Ontology are indicated with the acronym "SO" (e.g., SO:Action and SO:Event).

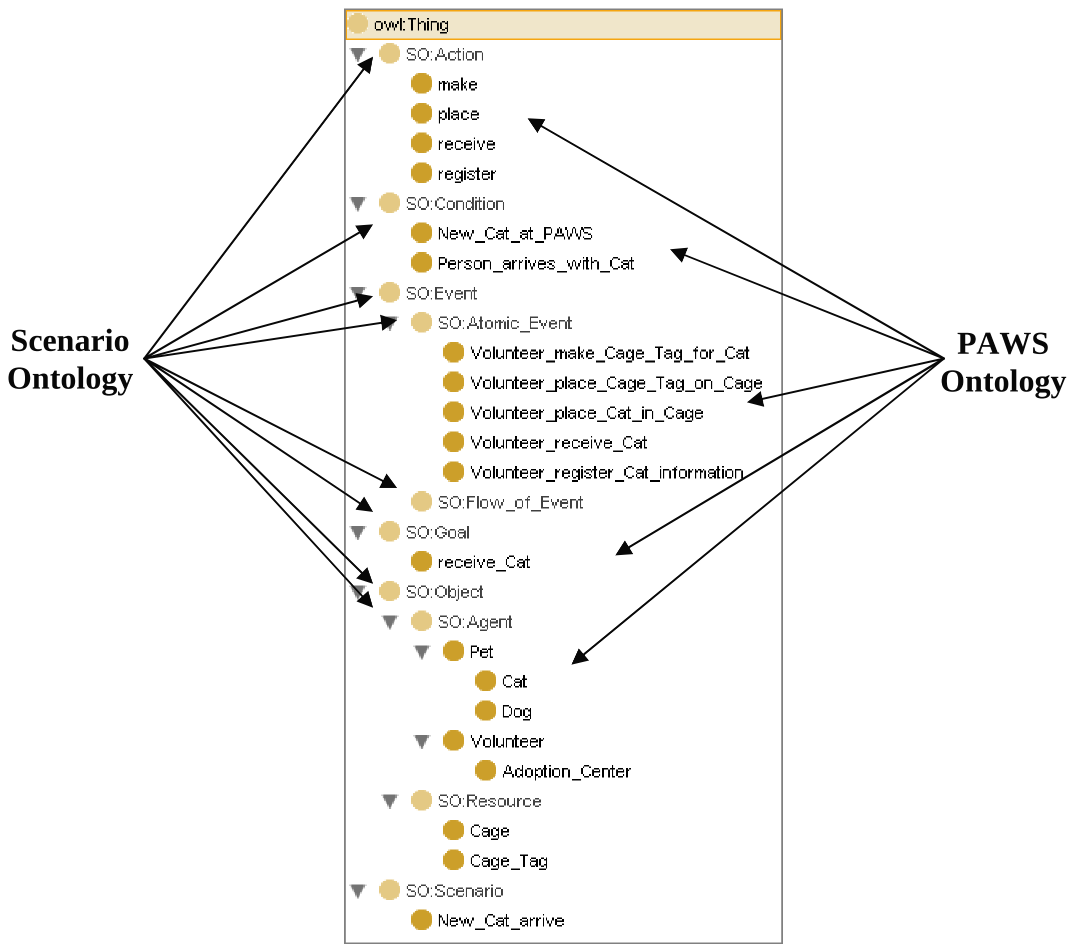

Figure 37: PAWS ontology (dark dot) mapped into the scenario ontology (light dot)

With the metamodel ontology in place, we can represent the concepts and relationships regarding the animal shelter domain. For instance, Figure 38 shows the formalization of the event labeled “Volunteer_place_Cat_in_Cage”. Following the example of clause patterns, we express the event with *subject* = *Volunteer*, *verb* = *place*, *objectDirect* = *Cat*, and *objectIndirect* = *Cage*. The properties *subject*, *objectDirect*, and *objectIndirect* are restricted with *allValuesFrom* to enforce that only instances of the concepts defined can be part of the event. *Cardinality 1* is used to enforce that for one event; only one *Volunteer* can place only one *Cat* in only one *Cage*. The concept *place*, however, is restricted with *hasValue*, which means that a specific value is being assigned to the property *verb*. So, the restriction reads, only one action is permitted in this event and the action must be *place*. The property *predecessor* is also restricted with *allValuesFrom* to enforce that an instance of this event occurs after an instance of an event labeled “Volunteer_receive_Cat” has occurred. Additional restrictions (i.e., axioms) would be required to ensure, for example, that the cat received is the same cat being placed in a cage.

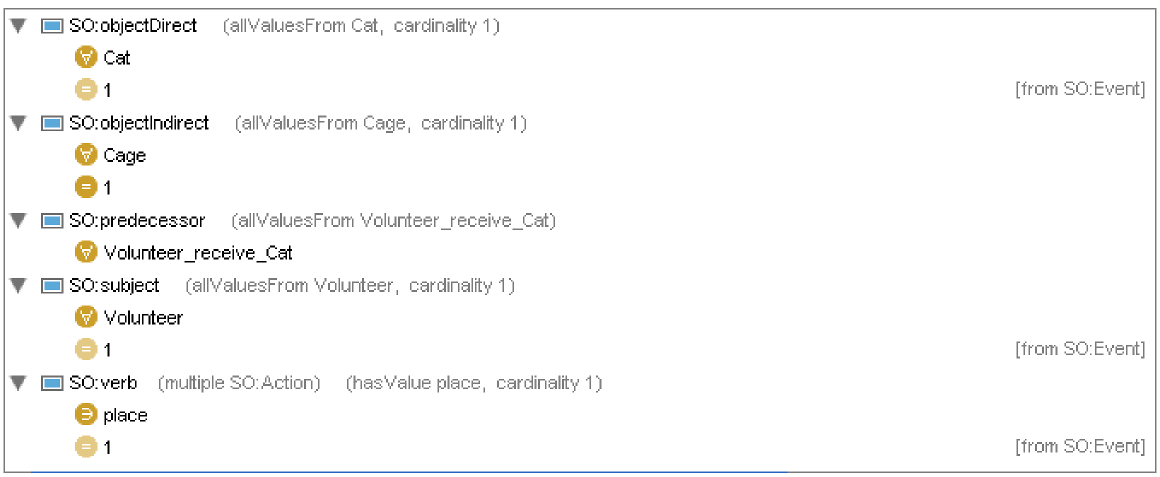

Figure 38: Definition of an atomic event for when the volunteer places cat in cage

An instance of an event labeled “Volunteer_place_Cat_in_Cage” is presented in Figure 39. It shows that we can attribute instance values to all properties, except to the property *verb*, which has been restricted to not allow instances. From this instance, humans should be able to understand roughly that a volunteer named Joseph placed a cat named Connor in a cage identified by the number 2. Machines, however, would need more details about the meaning of the action *place*. Representing the meaning of place is out of the scope of this experiment.

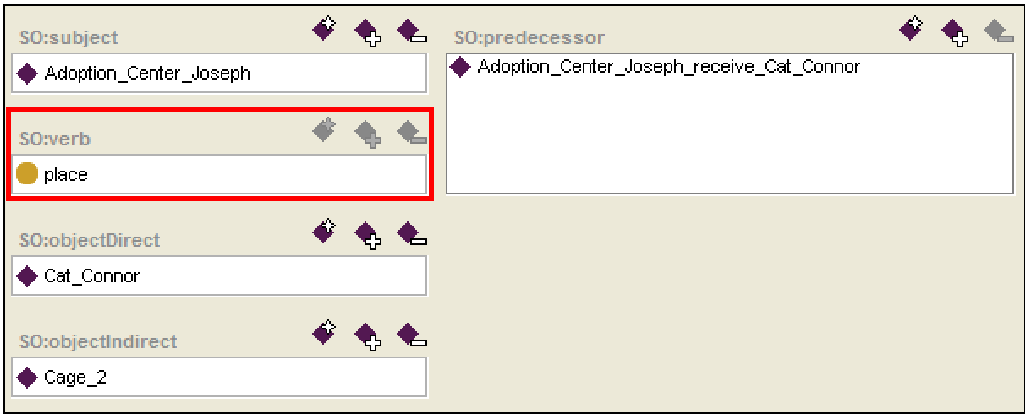

Figure 39: Instance of an atomic event for when a volunteer places cat in a cage

After creating the events, we can represent a sample scenario about a new cat arriving at PAWS. The goal of the scenario is to receive a cat and the scenario is triggered when someone brings a homeless cat to the animal shelter. A volunteer (e.g., adoption center representative) receives the cat and places the animal in a cage. Then, the volunteer register some information about the cat, and makes a tag that will be displayed on the cage. Figure 40 shows the properties and restrictions used to define this scenario. It is not the intention for this experiment to cover all restrictions involved with a scenario. So, additional restrictions would be required to ensure, for example, that all the events in the scenario are related the same cat.

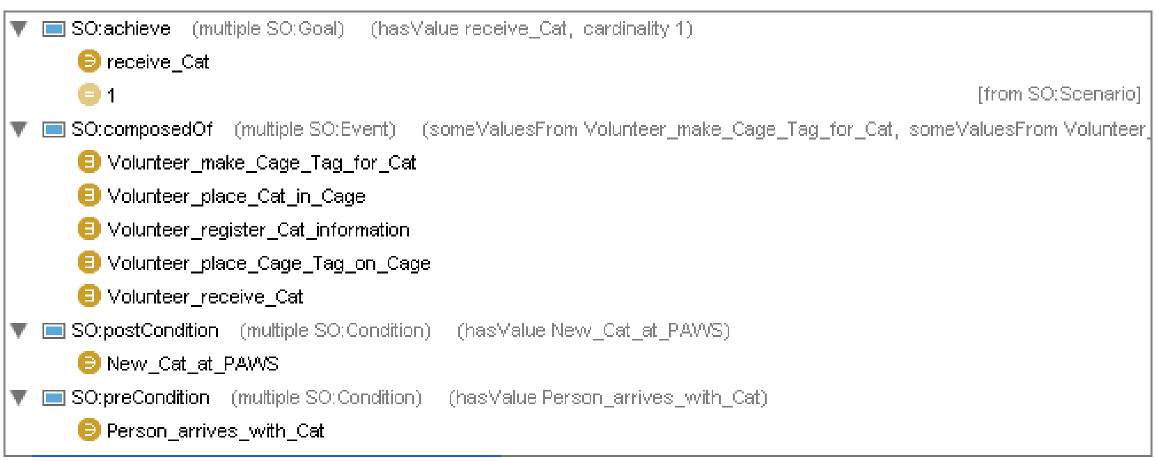

Figure 40: Definition of a scenario for when a new cat arrives at PAWS

Next, we present our comments on the use of scenarios for building such ontologies, especially with regards to the issues raised in Chapter 3 (i.e., knowledge acquisition, procedural knowledge, temporal relations, and metamodel).

6.4.3 Results and remarks

The use of a metamodel ontology based on scenarios has been confirmed as a valuable mechanism to build domain ontologies suitable to IS modeling. The metamodel provides a frame and view of the domain as a system, which helps domain experts and ontology designers to identify relevant domain concepts to be represented. This approach could augment the methodological steps of some of the methodologies investigated in this research, especially because the use of scenarios as promoted here addresses all four issues raised in Chapter 3.

Using a metamodel ontology based on scenarios provides a frame to view the domain as a system with goals and events. The metamodel also provides a guide for building domain ontologies, where concepts from the domain are mapped into concepts of the metamodel. Identifying the relevant domain concepts to be represented by the ontology can begin by finding scenarios within the domain. For each scenario, we find the sequence of events related to it, and for each event, we identify the specific concepts involved with the event. The relationships between concepts, in this case, are represented mainly by the actions of each event. Nevertheless, the domain ontology is still flexible in terms of creating concepts and relationships that are not mapped into the metamodel ontology.

As scenarios have specific goals, we could also start by looking for the specific goals within the domain. For instance, the main goal of the animal shelter is to find homes for the pets (i.e., to facilitate the adoption of a pet). This high-level goal is based on two sub-goals: to have a pet available for adoption and to have an adopter interested in adopting a pet. Each of these goals can be broken down into more specific goals, and consequently into more specific events that would be performed to achieve more specific goals.

In this experiment, we used task analysis, as proposed by Rosson & Carroll (2001), to identify the main events in the animal shelter domain. The resulting clusters of tasks turned into our scenarios, and the tasks turned into our events. Constraints between tasks, such as repetition and condition, have not been addressed by this experiment.

Scenarios have extensively been used to acquire knowledge used to build domain ontologies (Gandon, 2002; Giboni *et al.*, 2002; Grüninger & Fox, 1995; Lee, 2006). However, they are discarded once the harvest of concepts and relationships is completed. By using the components of scenarios as ontological constructs of a metamodel ontology, and mapping the domain concepts into the metamodel concepts, we see that the process of building scenarios about a domain is also the process of building domain ontologies. For instance, we can add the concept of an *adopter* to the domain ontology, and place it under the taxonomic structure of an *agent*. This way, *adopter* is both a concept in the domain ontology and an element of a scenario. We argue that the domain ontology created is enhanced by including knowledge about how the domain works.

As Rosson & Carroll (2002) explain, scenarios are stories about people performing some activities to achieve goals. Indeed, from the acquisition to the representation of domain knowledge, the most notable advantage of using scenarios to build ontologies for Information Systems is to identify and represent the sequence of events required to achieve the specific goal of a scenario. For Information Systems, it is important to identify and represent how the events are connected to each other, and especially what their dependencies are.

Inspired by how a Gantt Chart connects all its tasks, we included a property called *predecessor* in the concept *Event*. The property in each event provides information about the ordering in which the events are expected to be performed, and how the events depend on each other. In some cases, events within a scenario may be dependent on the completion of events from other scenarios. For instance, the event of a cat coordinator sending out the weekly cat census triggers both the scenario for *updating the website with cats for adoption*, and the scenario for *taking pictures of new cats*. However, at some point, adding information about a cat into the website will depend on receiving pictures of the cat.

By identifying the dependencies between events, we should be able to infer the temporality of the events in terms of, for example, Allen's (1983) interval algebra. However, this experiment does not account for all possible intervals. For instance, in the scenario of a new cat arriving at the shelter, the events *registering information about the cat* and *placing the cat in a cage* can assume different temporal intervals, such as placing before registering, registering before placing, or placing during the registration process.

The temporality of events can be used to predict some behaviors of the system. For instance, if a new cat and an adopter arrive at the shelter at the same time, and the adopter chooses to adopt the cat, the time for the adopter to get the cat would take about a week because the cat must undergo health inspection, be examined by a veterinarian, and other events before being ready for adoption.

In terms of defining constraints about the domain, the experiment was limited to the basic restrictions offered by the ontology editor Protégé. In particular, we used the restrictions shown in Figure 41. More detailed constraints and rules could be defined with the help of some logic representation such as first-order logic. However, exploring the constraints in detail is beyond the scope of this proof-of-concept experiment.

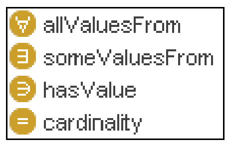

Figure 41: Restrictions used with the metamodel and domain ontology

We consider that using scenarios to build ontologies for IS has several advantages that could improve the steps of existing methodologies. In Table 13, we showed the ranking of the methodologies analyzed with regard to their contribution to the purpose of this research (i.e., quality assessment). From that table, we took the top five methodologies to compare their approaches with the `Scenario-based` approach discussed in this chapter. The points of comparison are the four issues identified in this research. Following, we summarize the comparison between the approaches (see Table 14).

Table 14: Comparison between the scenario-based approach and the methodological approaches

	Knowledge Acquisition	Procedural Knowledge	Temporal Relations	Metamodel
`O4IS`	Scenarios written in natural language or use case diagrams	SAR Functional relationships: Actor/Object-verb-Actor/Object/ Action and verb patterns	SAR Temporal relationships: Follow/Precede, Requires	Domain Ontology, ECA Rule Ontology, Performative Verbs Ontology, Verb Phrase Ontology
`DKAP`	Create statement pool and other source documents	Identify relevant design activities	N/A	Site-specific, practice, and domain ontology
`Grüninger & Fox`	Motivating scenarios and competency questions	Informal Competency Questions related to Planning and Scheduling	Information Competency Questions related to Temporal Projection, and predicates Poss, Do, do, holds, actual, and occurs	Activity Ontology and Organization Ontology
`Heuristic -Based`	Use cases	Pre-requisite constraints and domain dependencies.	Temporal constraint, mutually inclusive/exclusive	Break domain into sub-ontologies
`Bowman`	Problem-solving as tasks, questions and answers	Task reduction	N/A	N/A
`Scenario-Based`	Identify Scenarios and events decomposition	Sequence of events defined with the components of a scenario.	Defined through the property *predecessor*	Scenario Ontology and Domain Ontology

With regard to knowledge acquisition, `O4IS`, `Grüninger & Fox`, `DKAP` and `Bowman` rely on textual narrative to capture knowledge from the domain. `O4IS` also promotes the use of use cases, which is also employed by the `Heuristic-based`. Use cases for these approaches are represented as UML diagrams, which can be transformed into ontologies. The `Scenario-based` approach, however, uses the structure of scenarios as an ontology where the domain knowledge is mapped into. In this case, no transformation is necessary because the content of scenarios is already in the form of ontologies.

Identifying relevant concepts from the domain is covered by all six approaches. Nevertheless, because of the structure of use cases and scenarios, `O4IS`, `Heuristic-based` and `Scenario-based` provide substantial systematic ways for capturing relevant domain knowledge. The experiment presented in this chapter, for example, used task analysis to identify and decompose complex events. Similarly, `Bowman` employs task reduction to decompose problems; however, the content is structured as questions and answers, which later must be transformed into ontologies.

With regard to procedural knowledge, `Scenario-based` already contains the knowledge captured as an ontology, whereas the other approaches will undergo some transformation from the textual scenarios to ontologies. `O4IS` and `Scenario-based` present distinct structures to represent procedural knowledge, and both use similar template to express events (e.g., subject-verb-object). However, `O4IS` includes a more detailed description of the meaning of the actions, by using verb patterns. `Heuristic-based`, `Grüninger & Fox`, `DKAP`, and `Bowman` present some description of what to do to capture procedural knowledge, but with little description of how to do it.

With regard to temporal relations, `Scenario-based` and `O4IS` have similar constructs to represent temporality between events in the domain. `Grüninger & Fox` present some constructs to describe temporality, while `Heuristic-based` lists steps to represent temporal constraints without much detail of how to do it and what structure to use. `Heuristic-based`, however, is the only approach addressing concurrency between events through mutually inclusive/exclusive constraints.

With regard to metamodels, `Grüninger & Fox`, `Heuristic-based` and `DKAP` use different types of ontologies, but not metamodel ontologies. In this case, different parts of the domain are stored into different ontologies. Both `Scenario-based` and `O4IS` rely on a high-level ontology to frame and build domain ontologies. `Scenario-based` uses scenario ontology as a metamodel, and `O4IS` uses ECA rule ontology, Performative Verbs Ontology, and Verb Phrase Ontology. As of now, `Scenario-based` is an experiment that does not provide support for defining verbs.

We argue that the proposed `Scenario-based` is not directly competing with the other approaches. Instead, we see it as an important approach that can improve the steps of existing methodologies. In particular, we suggest the adoption of the `Scenario-based` approach to support knowledge acquisition. We argue that the use of scenarios in the process of building domain ontologies can reduce the number of steps to build such ontologies, and has the potential to provide formalized knowledge about the domain.

With `Scenario-based` approach, the process of representing scenarios is already part of the process of creating domain ontologies, which means fewer steps to build an ontology and no need to transform, for example, scenarios into ontologies. The formalized representation with ontologies should allow machines to handle the content of scenario, especially with regard to knowledge reuse, and automated views of the systems. For instance, an application handling the `Scenario-based` approach should be able to produce, with the knowledge from the ontologies, a view of a specific scenario with its tasks and agents, or a view of all scenarios that include a specific agent.

The `Scenario-based` approach is a process at the knowledge acquisition level that captures domain knowledge including concepts, relations, procedural knowledge, and temporal relations. This approach should be for the most part a seamless transition with regard to what type of information to represent, yet it should be flexible to allow representation of individual types of information. For example, creating a list of terms and then using the terms to create scenarios. In this case, to include the terms into a scenario means to map the terms into the respective metamodel constructs. Finally, we envision the content of an ontology being transformed into IS components (Kishore *et al.*, 2004) because the ontology contains information about how the system works, and what information is needed and when. That is, the ontology should be able to provide information about the application programs, the interfaces and the information resources.

Compared to the other approaches, the `Scenario-based` approach seems to improve the process of knowledge acquisition. However, with regard to procedural knowledge and metamodel, the `O4IS` approach has a more detailed structure; and with regard to temporal relations, `O4IS` and `Scenario-based` contain a similar structure. The idea of using scenarios resulted from our analysis of existing methodologies, where we observed a frequent use of scenarios in the initial steps of some methodologies. Thus, we were interested in finding out how scenarios could be used to improve existing approaches. Our suggestion is to use the components of scenarios as ontological constructs to enhance the domain ontology by including a representation of procedural knowledge.

To conclude this chapter, it is important to mention that the use of a Scenario Ontology to frame knowledge acquisition has already been proposed by Yu-N & Abidi (2000a, 2000b); however, their approach does not cover the creation of domain ontologies. Instead, it creates instances of scenarios, which are represented with XML. According to Yu-N & Abidi, the scenario instances could be translated, with the help of the scenario ontology, into other representation languages, such as Prolog. Another use of scenario ontology includes selecting concepts from scenarios written as textual narratives, and adding the concepts into an ontology called Scenario Ontology (Gotts & Polhill, 2009; Polhill & Ziervogel, 2006).

6.5 Summary

In this chapter we addressed the major research questions of this research and presented a proof-of-concept experiment to illustrate the use of scenarios in the process of building ontologies suitable for IS modeling. The experiment showed that a scenario-based approach covers the issues identified in this research and offers improvements over existing methodologies. Next, we conclude the research with a summary of the research accomplishments and contributions, as well as future research directions.

Chapter 7.
CONCLUSIONS

This research focused on identifying principled guidelines for building ontologies to be used in Information Systems modeling. We searched major bibliographic databases and selected 30 methodologies to investigate their approaches to build ontologies. The analysis of the methodologies was formulated around the core components of an ontology, and the issues related to the process of building such ontologies. The research presented the main features of the methodologies that can contribute to the building of ontologies for IS. The research proposed a scenario-based approach that addresses the issues found. We argue that scenarios can be used to enhance the methodological approaches of existing methodologies.

7.1 Summary of the dissertation

This research targeted the area of Ontology-Driven Information Systems, where ontologies play a central role both at development time and at run time of Information Systems (Fonseca, 2007; Guarino, 1998; Uschold, 2008). In particular, the research was interested in the process of building domain ontologies for IS modeling.

The main motivation to pursuing the research was the fact that: (1) researchers have not yet produced comprehensive guidelines for building ontologies for IS, (2) 60% of participants of a recent survey (Cardoso, 2007) reported that they did not use any methodology to build their ontologies, (3) ontology engineering is still considered an art, rather than and engineering activity, and (4) the results of our preliminary research on building an ontology for a given domain revealed four important issues to be considered when building ontologies for IS.

The issues identified are related to:

- *Metamodels*: a high-level structure that provides a framed view of the domain, and guides the construction of domain ontologies and increases the semantic for understanding the domain.

- *Procedural knowledge*: describes a set of tasks that need to be performed for achieving a goal.
- *Temporal relations*: represent the chronological arrangement of the tasks and their dependencies.
- *Knowledge acquisition*: relates to a systematic approach for capturing relevant domain knowledge to be represented by the ontology.

Based on the concerns above, we set up a research to investigate existing methodologies that could provide methodological approaches to overcome the issues raised. The research adopted a formal method, called Systematic Reviews (Kitchenham, 2004; Petticrew & Roberts, 2006; Torgerson, 2003), to conduct a literature review of methodologies to build ontologies. We searched 2025 publications from major bibliographic databases (i.e., ACM, IEEE, Springer, Elsevier, Web of Science, and Proquest) from which we selected 30 methodologies to investigate. The methodologies were analyzed with regard to the twelve categories that we developed to cover mainly the core components of an ontology (i.e., concepts, properties, relationships and axioms), and the four issues identified from our preliminary research. Lastly, we discussed the methodological features that are relevant to the process of building ontologies for Information Systems.

The frequent use of scenarios in the initial steps of the methodologies motivated us to further investigate its use in building ontologies. We proposed to use the components of a scenario (e.g., agents, goals, and events) as the ontological constructs of a metamodel ontology. So, instead of using, for example, traditional textual narratives to describe scenarios, we use formalized ontologies. To illustrate the use of scenarios in the building of domain ontologies, we created a proof-of-concept experiment. The experiment successfully showed that viewing the domain as a scenario helps acquiring and representing relevant domain knowledge to be used in IS modeling, especially with regards to procedural knowledge and temporal relations.

7.2 Results and major findings

This research presents significant contributions to researchers and practitioners in the area of Ontology-Driven Information Systems. The research has identified four issues related to the process of building ontologies for IS and has confirmed that the issues indeed exist among the methodologies analyzed. In addition, as a result of the analysis of the methodologies, we identified that axiomatization has emerged as an important issue in the process of building ontologies. An axiom is considered a core component of an ontology. However, steps to defining and representing it are not clearly described within the methodologies, which leave domain experts and ontology designers with the responsibility to figure out how to identify and represent constraints of the domain. In terms of Ontology-Driven Information Systems, the use of axioms is even more crucial if we want ontologies to really take a central position in Information Systems.

The timeline of the selected methodologies (see Figure 22) raises some concerns regarding the number of methodologies created over a period of almost three decades. It seems that new methodologies are being proposed without much consideration of the lessons learned from previous approaches. Uschold (1996) has already proposed to combine methodological approaches "into a coherent framework which might be in the form of a handbook, which would provide useful guidance for anyone wishing to build an ontology" (p.3). Our research partially fills this gap as we uncover practices that can be used to build ontologies for IS.

The lack of principled methodologies (Guarino & Welty, 2000) and the limitted use of methodologies to build ontologies, as shown in Cardoso's (2007) survey, can be some of the reasons why ontology engineering is still considered an art. For that to change, we argue for the need of sound methodological approaches to build ontologies. A design methodology should be to ontology what a research methodology is to science. No scientific research is considered trustworthy if it is not supported by well-defined research methodologies. Similarly, no ontology should be considered trustworthy if it is not supported by well-defined methodologies to build ontologies.

We expect that uncovering other methodologies, beyond the top nine presented in Cardoso's survey, and discussing their relevant features to build ontologies for IS, has shed light on the improvements of existing methodologies or the design of future ones. We argue that the review of methodologies has produced valuable results for researchers and practitioners by uncovering specific approaches and methodologies to build ontologies for IS. In addition, we consider that the scenario-based approach, demonstrated in the proof-of-concept experiment in Chapter 6, can be profitable in the process of building ontologies for IS, and a great enhancement to existing methodologies. The scenario approach fruitfully addresses the four issues raised by this research.

The results of this research help to enhance not only the process of building ontologies but also the quality of the ontologies created, as they will include more details about the domain being represented. However, there is much still to be investigated within the realm of Ontology-Driven Information Systems. This area "is in the early stages of becoming a practical reality" as it depends on the development and improvement of important technologies in the area of Ontology Engineering, Knowledge Representation, and Information Systems Analysis and Design (Uschold, 2008, p.16).

7.3 Future work

Building upon the results of this work, we plan to continue investigating the creation and use of ontology in Information Systems (see Figure 42) with the following research initiatives.

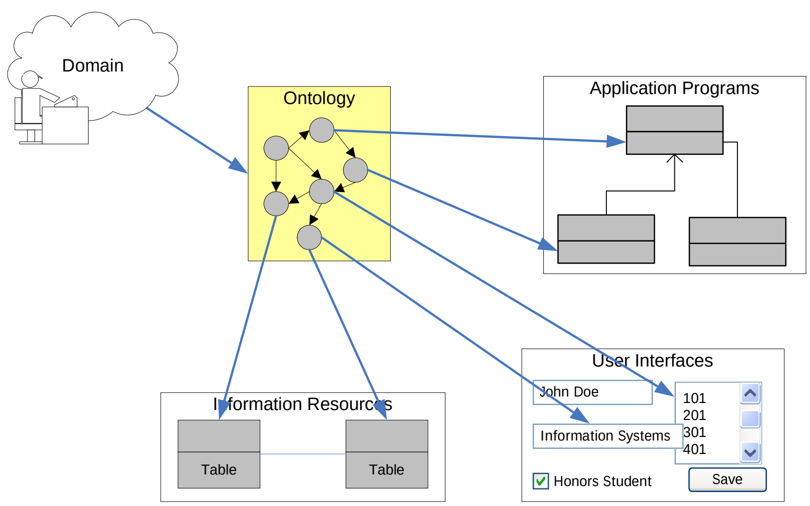

Figure 42: View of ontology playing a central role in Information Systems

7.3.1 The creation of metamodel ontology based on scenarios

Motivating scenarios was frequently used in the initial steps of the methodologies as a means to identify domain concepts and their relationships. However, the scenario itself was not represented in the ontology. Our proof-of-concept experiment has shown that a scenario can be used as a metamodel ontology to guide the construction of domain ontologies, and can provide a richer semantic for machines to understand content meaning. This research initiative will further explore the components of a scenario that can be used to represent temporality and other controls (e.g., repetition and condition) between events.

7.3.2 Structured English for ontology design

To allow end-users (i.e., non-ontologists) to build their own ontologies without the burden of learning the underpinnings of ontology engineering, mechanisms are needed that would help them to describe their domains in a way that is similar to how they think and communicate. Our experiment with declarative clauses has shown the feasibility of building domain ontologies that can be read by both humans and machines. This research initiative will explore declarative clauses in more details to identify opportunities to represent domain knowledge with ontologies.

7.3.3 Transforming ontology into IS components

A truly Ontology-Driven Information System is a system that operates in accordance with the specifications of the ontology. In this research initiative, we will focus on mechanisms that can automatically transform the content of an ontology into IS components, so that the components would reflect the knowledge embedded in the ontology. We also intend to investigate if the time spent building an ontology that will be transformed into a system would pay off when compared with the time spent directly designing the system.

7.3.4 Extensions to other domains

Capturing, representing and reusing domain knowledge are common activities for several domains. In this research initiative, we will investigate how the proposed scenario-based ontology can be used in engineering and other non-IS domains to create ontologies that represent their domain knowledge.

Bibliography

Abou-Zeid, E. S. (2003). What can ontologists learn from knowledge management? *Journal Of Computer Information Systems, 43*(3), pp.109-117.

Achour, C. B. (1998). *Writing and Correcting Textual Scenarios for System design.* In Proceedings of the Proceedings of the Natural Language and Information Systems - DEXA Workshop, Vienna, Austria, pp.166-170.

Ahmed, S., Kim, S., & Wallace, K. M. (2007). A methodology for creating ontologies for engineering design. *Journal Of Computing And Information Science In Engineering, 7*(2), pp.132-140.

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly, 25*(1), pp.107-136.

Alexander, I. (2004). Introduction: Scenarios in System Development. In I. F. Alexander & N. Maiden (Eds.), *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle* (pp.3-24): John Wiley & Sons.

Alexander, I., & Maiden, N. (Eds.). (2004). *Scenario, Stories, Use Cases: Through the Systems Development Life-Cycle*: John Wiley and Sons, pp.Pages.

Allee, V. (1997). *The Knowledge Evolution: Expanding Organizational Intelligence*: Butterworth-Heinemann.

Allen, J. F. (1983). Maintaining Knowledge about temporal intervals. *Communications of the ACM, 26*(11), pp.832-843.

Bachimont, B., Isaac, A., & Troncy, R. (2002, October 1-4). *Semantic Commitment for Designing Ontologies: A Proposal.* In Proceedings of the 13th International Conference, EKAW 2002, Siguenza, Spain, pp.114-121.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in Software Engineering: A systematic literature review. *Information and Software Technology, 50*, pp.860-878.

Benslimane, D., Arara, A., & Yetongnon, k. (2003). *Two approaches for ontologies building: From-scratch and From existing data sources.* In Proceedings of the 2003 International Conference on Information Systems and Engineering, pp.189-193.

Borst, W. N. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse.* Doctoral Dissertation, University of Twente, Ensched - The Netherlands.

Bowman, M. (2002). *A methodology for modeling expert knowledge that supports teaching-based development of agents.* Doctoral Dissertation, George Mason University, Fairfax, Virginia - USA.

Breitman, K., & Leite, J. C. S. (2003, 8-12 Sept). *Ontology as a requirement engineering product.* In Proceedings of the Eleventh IEEE International Requirements Engineering Conference, Monterey Bay, California, pp.309–319.

Breitman, K. K., Casanova, M. A., & Truszkowski, W. (2007). Methods for Ontology Development. In *Semantic Web: Concepts, Technologies and Applications* (pp.155-173): Springer London.

Brusa, G., Caliusco, M. L., & Chiotti, O. (2006). A process for building a domain ontology: an experience in developing a government budgetary ontology,

Proceedings of the second Australasian workshop on Advances in ontologies - Volume 72. Hobart, Australia: Australian Computer Society, Inc.
Bubenko, J. A. (1979). *On the Role of 'Understanding Models' in conceptual schema design.* In Proceedings of the Fifth International Conference on Very Large Data Bases, Rio De Janeiro, Brazil, pp.129-139.
Buchanan, B., Barstow, D., Bechtal, R., Bennet, J., Clancey, W., Kulikowski, C., *et al.* (1983). Constructing an Expert System. In F. Hayes-Roth, D. Waterman & B. Lenat (Eds.), *Building Expert Systems* (pp.127-167). Reading, MA: Addison-Wesley.
Bunge, M. (1977). *Treatise on Basic Philosophy: Volume 3, The Furniture of the World*: D. Reidel Publishing.
Cardoso, J. (2007). The Semantic Web Vision: Where are We? *IEEE Intelligent Systems, 22*(5), pp.84-88.
Carroll, J. M. (2000). *Making Use: Scenario-Based Design of Human Computer Interactions*. Cambridge, MA: MIT Press.
Carroll, J. M. (2006). Scenario-Based Design. In W. Karwowski (Ed.), *International Encyclopedia of Ergonomics and Human Factors* (2nd ed., Vol. 1, pp.160-163): CRC Press.
Carroll, J. M. (Ed.). (1995). *Scenario-based Design: Envision Work and Technology in System Development*. New York, NY: John Wiley and Sons, pp.Pages.
Chen, L.-L., & Chan, C. W. (2001). *Ontology Design and Its Application in the Petroleum Remediation Domain.* In Proceedings of the Four Workshops held at PRICAI 2000, Melbourne, Australia, pp.16-23.
Chen, P. S. S. (1976). The Entity-Relationship Model: Towards a Unified View of Data. *ACM Transactions on Database Systems, 1*(1), pp.9-36.
Cooper, H., & Hedges, L. V. (2009). Research Synthesis as a Scientific Process. In H. Cooper, L. V. Hedges & J. C. Valentine (Eds.), *THe Handbook of Research Synthesis and Meta-Analysis* (2nd ed., pp.3-16). New York: Russel Sage Foundation.
Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies: where is their meeting point? *Data & Knowledge Engineering, 46*(1), pp.41-64.
Cordingley, E. S. B. (1989). Knowledge Elicitation Technique for Knowledge-Based Systems. In D. Diaper (Ed.), *Knowledge Elicitation principles, techniques and applications* (pp.89-175): Halsted Press.
Davies, I., Green, P., Milton, S., & Rosemann, M. (2005). Analyzing and Comparing Ontologies with Meta-Models. In J. Krogs, T. Halpin & K. Siau (Eds.), *Information Modeling Methos and Methodologies* (pp.1-16): Idea Group.
Davis, R., Shrobe, H., & Szolovits, P. (1993). What is a Knowledge Representation? *AI Magazine, 14*(1), pp.17-33.
Dennis, A., Wixom, B. H., & Roth, R. M. (2005). *Systems Analysis Design* (Third ed.): John Wiley & Sons.
Devedzic, V. (2002). Understanding Ontological Engineering. *Communications of the ACM, 45*(4), pp.136-144.

Diaper, D. (1989). Designing Expert Systems - From Dan to Beersheba. In D. Diaper (Ed.), *Knowledge Elicitation principles, techniques and applications* (Vol. 1, pp.15-46). New York, NY: Springer-Verlag New York.

Dong-Soon, K., Suk-Hyung, H., & Hong-Gee, K. (2007). *Concept Analysis of OWL Ontology Based on the Context Family Model.* In Proceedings of the 2007 International Conference on Convergence Information Technology (ICCIT 2007), pp.896-901.

Evermann, J. (2004). Thinking Ontologically; Conceptual vs. Design Models in UML. In P. Green & M. Rosemann (Eds.), *Business Systems Analysis with Ontologies* (pp.82-104). Hershey, PA: Idea Group publishing.

Falbo, R. A., Guizzardi, G., & Duarte, K. (2002). *An ontological approach to domain engineering.* In Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE'02), Ischia, Italy.

Falbo, R. d. A., Menezes, C. S. d., & Rocha, A. R. (1998). *A Systematic Approach for Building Ontologies.* In Proceedings of the 6th Ibero-American Conference on AI: Progress in Artificial Intelligence (IBERAMIA'98), Lisbon, Portugal, pp.349-360.

Falkenberg, E. D., Hesse, W., Lindgreen, P., Nilsson, B. E., Oei, J. L. H., Rolland, C., *et al.* (1998). *A Framework of Information Systems Concepts: The FRISCO Report*: International Federation for Information Processing (IFIP).

Fernandez-Lopez, M. (1999). *Overview of the methodologies for building ontologies.* In Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends, Stockholm, Sweden, pp.4.1-4.13.

Fernandez-Lopez, M., & Gómez-Pérez, A. (2002). Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review, 17*(2), pp.129-156.

Fernandez, M., Gómez-Pérez, A., & Juristo, N. (1997). *METHONTOLOGY: From Ontological Art Towards Ontological Engineering.* In Proceedings of the 1997 AAAI Spring Symposium, Menlo Park, California - USA, pp.33-40.

Fettke, P., & Loos, P. (2003). *Ontological Evaluation of Reference Models Using the Bunge-Wand-Weber Model.* In Proceedings of the Ninth Americas Conference on Information Systems (AMCIS'2003), Tampa, Florida, pp.2944-2955.

Fonseca, F. (2007). The Double Role of Ontologies in Information Science Research. *Journal of the American Society for Information Science and Technology, 58*(6), pp.786-793.

Fonseca, F., & Martin, J. (2007). Learning the Differences between Ontologies and Conceptual Schemas through Ontology-Driven Information Systems. *Journal of the Association for Information Systems, 8*(2), pp.129-142.

Fonseca, F. T., & Martin, J. E. (2005). Toward an Alternative Notion of Information Systems Ontologies: Information Engineering as Hermeneutic Enterprise. *Journal of the American Society for Information Science and Technology, 56*(1), pp.46-57.

Gandon, F. (2002). Ontology Engineering: A Survey and a Return on Experience. RR-4396. from ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-4396.pdf

Gangemi, A., Guarino, N., Masolo, C., & Oltramari, A. (2003). Sweetening WorldNet with DOLCE. *AI Magazine, 24*(3), pp.13-24.

Gavrilova, T., & Laird, D. (2005, August 25-27, 2005). *Practical Design of Business Enterprise Ontologies.* In Proceedings of the 1st International IFIP/WG12.5

Working Conference on Industrial Applications of Semantic Web, Jyvaskyla, Finland, pp.65-81.
Gavrilova, T., Voinov, A., & Vasilyeva, E. (1999). *Visual knowledge engineering as a cognitive tool.* In Proceedings of the International Work-Conference on Artificial and Natural Neural Networks, IWANN'99, Alicante, Spain, pp.750-758.
Giboni, A., Gandon, F., Corby, O., & Dieng, R. (2002). *Assessment of Ontology-based Tools: A Step Towards Systemizing the Scenario Approach.* In Proceedings of the EON2002. Evaluation of Ontology-based Tools Workshop in the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002, Siguenza, Spain.
Go, K., & Carroll, J. M. (2004). The Blind Men and the Elephant: Views of Scenario-Based System Design. *Interactions, 11*(6), pp.44-53.
Gómez-Pérez, A. (1998). Knowledge sharing and reuse. In J. Liebowitz (Ed.), *Handbook of Applied Expert Systems* (pp.10-11 to 10-36). Boca Raton, Florida - USA: CRC Press.
Gómez-Pérez, A. (1999). Ontological Engineering: A state of the Art. *Expert Update, 2*(3), pp.33-43.
Gómez-Pérez, A., Fernandez-Lopez, M., & Corcho, O. (2004). *Ontological Engineering: with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web* (1st ed.): Springer Berlin Heidelberg.
Gómez-Pérez, A., Fernendez, M., & Vicente, A. J. d. (1996). *Towards a method to conceptualize domain ontologies.* In Proceedings of the Workshop on Ontological Engineering. European Conference of Artificial Intelligence (ECAI'96), Budapest, Hungary, pp.41-52.
Gotts, N. M., & Polhill, J. G. (2009). Narrative Scenarios, Mediating Formalisms, and the Agent-Based Simulation of Land Use Change. In *Epistemological Aspects of Computer Simulation in the Social Sciences* (Vol. 5466, pp.99-116): Springer Berlin / Heidelberg.
Green, P., & Rosemann, M. (Eds.). (2005). *Business Systems Analysis with Ontologies*. Hershey, PA: Idea Group Inc, pp.Pages.
Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*, pp.199-220.
Grüninger, M., & Fox, M. S. (1994). *The role of competency questions in enterprise engineering.* In Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice, Trondhein, Norway, pp.2-31.
Grüninger, M., & Fox, M. S. (1995). *Methodology for the design and evaluation of ontologies.* In Proceedings of the IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, pp.6.1–6.10.
Guarino, N. (1998). *Formal Ontology in Information Systems.* In Proceedings of the Formal Ontology in Information Systems (FOIS'98), Trento, Italy, pp.3-15.
Guarino, N., & Giaretta, P. (1995). Ontology and Knowledge Bases: Towards a Terminology Clarification. In N. J. I. Mars (Ed.), *Towards Very Large Knolwedge Bases*. Amsterdam: IOS Press.
Guarino, N., & Guizzardi, G. (2006). In defense of Ontological Foundations for Conceptual Modeling. *Scandinavian Journal of Information Systems, 18*(1), pp.115-126.

Guarino, N., & Welty, C. (2000). *A Formal Ontology of Properties.* In Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW'2000), pp.97-112.

Guizzardi, G. (2005). *Ontological Foundations for Structural Conceptual Models.* Doctoral Dissertation, University of Twente, Twente, The Netherlands.

Guizzardi, G., & Wagner, G. (2004). Some Applications of a Unified Foundational Ontology in Business Modeling. In P. Green & M. Roseman (Eds.), *Business Systems Analysis with Ontologies* (pp.345-367). Hershey, PA: Idea Group Publishing.

Helbig, H. (2006). *Knowledge Representation and the Semantics of Natural Language* (1st ed.). New York, NY: Springer-Verlag Berlin Heidelberg.

Hertzum, M. (2003). Making Use of Scenarios: A Field Study of Conceptual Design. *International Journal of human-Computer Studies, 58*(2), pp.215-239.

Hou, C.-S. J., Musen, M. A., & Noy, N. F. (2005). EZPAL: Environment for composing constraint axioms by instantiating templates. *International Journal of Human-Computer Studies, 62,* pp.578-596.

Huang, C.-J., Trappey, A. J. C., & Wu, C.-Y. (2008). Develop a Formal Ontology Engineering Methodology for Technical Knowledge Definition in R&D Knowledge Management. In R. Curran, S.-Y. Chou & A. Trappey (Eds.), *Collaborative Product and Service Life Cycle Management for a Sustainable World Proceedings of the 15th ISPE International Conference on Concurrent Engineering (CE2008)* (pp.495-502): Springer London.

Hüsemann, B., & Vossen, G. (2005, December 7-9). *Ontology Engineering from a Database Perspective (Extended Abstract).* In Proceedings of the 10th Asian Computing Science Conference, ASIAN 2005, Kunming, China, pp.49-63.

IEEE. (1990). *IEEE Standard Glossary of Software Engineering Terminology* (No. Std 610.121990). New York: IEEE Computer Society.

Jarke, M., Bui, X. T., & Carroll, J. M. (1998). Scenario Management: An Interdisciplinary Approach. *Requirements Engineering, 3*(3-4), pp.154-173.

Jin, L., Keqing, H., Bing, L., Hao, C., & Liang, P. (2004). *A methodology for acquisition of software component attribute ontology.* In Proceedings of the The Fourth International Conference on Computer and Information Technology (CIT '04), pp.1058-1064.

Johson, J., & Henderson, A. (2002). Conceptual Models: Begin by Designing What to Design. *Interactions, 9*(1), pp.25-32.

Jong, T. d., & Ferguson-Hessler, M. G. M. (1996). Types and Qualities of Knowledge. *Educational Psychologist, 31*(2), pp.105-113.

Jung, E.-H., Cho, K.-M., Song, K.-H., Nam, S.-H., & Lee, S.-W. (2008). *Methodology of Topic Maps creation and Semantic Web for technological information search regarding injection-mold based on Collaboration Hub.* In Proceedings of the International Conference on Smart Manufacturing Application (ICSMA 2008), Gyeonggi-do, South Korea, pp.78-83.

Kishore, R., Sharman, R., & Ramesh, R. (2004). Computational Ontologies and Information Systems: I. Foundations. *Communications of the Association for Information Systems, 14*(8), pp.158-183.

Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews* (Joint Technical Report No. Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1).

Koenderink, N., van Assem, M., Hulzebos, J., Broekstra, J., & Top, J. (2008, December 8-11). *ROC: A Method for Proto-ontology Construction by Domain Experts.* In Proceedings of the 3rd Asian Semantic Web Conference (ASWC 2008) - The Semantic Web, Bangkok, Thailand, pp.152-166.

Kong, H., Hwang, M., & Kim, P. (2006). *Design of the automatic ontology building system about the specific domain knowledge.* In Proceedings of the The 8th International Conference Advanced Communication Technology (ICACT 2006), pp.1405-1408.

Kurtev, I. (2007). *Metamodel: Definitions of Structures or Ontological Commitments?* In Proceedings of the Workshop on Towers of Models, Zurich, Switzerland, pp.53-63.

Lee, J. (2006). The roles of scenario use in ontology development. *Knowledge and Process Management, 13*(4), pp.270-284.

Leite, J. C. S. d. P., hadad, G. D. S., Doorn, J. H., & Kaplan, G. N. (2000). A Scenario Construction Process. *Requirements Engineering, 5*, pp.38-61.

Lenat, D. B., & Guha, R. V. (1990). *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co.

Linehan, M. H. (2007). *Ontologies and Rules in Business Models.* In Proceedings of the Eleventh International IEEE EDOC Conference Workshop (EDOCW'07), pp.149-156.

Littell, J. H., Corcoran, J., & Pillai, V. (2008). *Systematic Reviews and Meta-Analysis*. New York: Oxford University Press.

Lyytinen, K. (2006). "Ontological Foundations of Conceptual Modeling" by Boris Wyssusek - A critical response. *Scandinavian Journal of Information Systems, 18*(1), pp.81-84.

Maedche, A. (2002). *Ontology Learning for the Semantic Web* (Vol. 665): Kluwer Academic Publishers.

McGuinness, D. L. (2001). Ontologies Come of Age. In D. Fensel, J. Hendler, H. Lieberman & W. Wahlster (Eds.), *The Semantic Web: Why, What, and How*: MIT Press.

Mealy, G. H. (1967). *Another Look at Data.* In Proceedings of the Fall Joint Computer Conference, Anaheim, California, pp.525-534.

Milton, N. R. (2007). *Knowledge Acquisition in Practice: A Step-by-Step Guide*: Springer-Verlag London.

Milton, S. K., & Kazmierczak, E. (2006). Ontology as Meta-Theory: A Perspective. *Scandinavian Journal of Information Systems, 18*(1), pp.85-94.

Mizoguchi, R. (2003). Tutorial on ontological engineering - Part 1: Introduction to Ontological Engineering. *New Generation Computing, 21*(4), pp.365-384, http://www.ei.sanken.osaka-u.ac.jp/pub/miz/Part361-pdf362.pdf.

Mylopoulos, J., Borgida, A., Jarke, M., & Koubarakis, M. (1990). Telos: Representing Knowledge About Information Systems. *Information Systems, 8*(4), pp.325-362.

Myloupolos, J. (1992). Conceptual Modeling and Telos. In P. Loucopoulos & R. Zicari (Eds.), *Conceptual modeling, databases, and CASE* (pp.49-68): Wiley.

Niles, I., & Pease, A. (2001). *Towards a standard upper ontology.* In Proceedings of the Formal Ontology in Information Systems (FOIS'01), Ogunquit, Maine, pp.2-9.

Noy, N. F., & McGuinness, D. L. (2001). Ontology Development 101: A Guide to creating your first Ontology. from http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf

Nunamaker, J. F., & Chen, M. (1990). *Systems development in information systems research.* In Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences, Kailua-Kona, HI, USA, pp.631-640.

Oberle, D. (2006). *Semantic Management of Middleware* (Vol. 1): Springer.

Peralta, D., Sofia Pinto, H., & Mamede, N. (2005a). Reusing a Time Ontology. In O. Camp, J. B. L. Filipe, S. Hammoudi & M. Piattin (Eds.), *Enterprise Information Systems V* (pp.241-248): Kluwer Academic Publishers.

Peralta, D., Sofia Pinto, H., & Mamede, N. (2005b). Reusing a Time Ontology. In *Enterprise Information Systems V* (pp.241).

Petticrew, M., & Roberts, H. (2006). *Systematic Reviews in the Social Sciences: A Practical Guide*: Blackwell Publishing.

Pinto, H. S., & Martins, J. P. (2004). Ontologies: How can They be Built? *Knowledge and Information Systems, 6*(4), pp.441-464.

Polhill, J. G., & Ziervogel, G. (2006, 28-30 June). Using ontologies with case studies: an end-user perspective on OWL. *Second International Conference on e-Social Science (NceSS 2006)*, from http://www.ncess.ac.uk/events/conference/2006/papers/papers/PolhillEndUserPerspectiveOnOWL.pdf

Qing, Y., QingXian, W., Yan, L., Qiang, W., & JunYong, L. (2007). *A New Methodology for Building Ontology Based on Reusing the Heterogeneous Ontologies.* In Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), Haikou, Hainan, China, pp.717-721.

Rolland, C., & Achour, C. B. (1998). Guiding the Construction of Textual Use Case Specifications. *Data & Knowledge Engineering Journal*, *25*(1-2), pp.125-160.

Rolland, C., & Prakash, N. (2000). From conceptual modelling to requirements engineering. *Annals of Software Engineering, 10*, pp.151-176.

Rolland, C., Souveyet, C., & Achour, C. B. (1998). Guidind goal modeling sing scenarios. *IEEE Transaction on Software Engineering*, *24*(12), pp.1055-1071.

Rosemann, M., & Wyssusek, B. (2005). *Enhancing the Expressiveness of the BWW Ontology.* In Proceedings of the Eleventh Americas Conference on Information Systems, Omaha, NE, USA, pp.2803-2810.

Rosson, M. B., & Carroll, J. M. (2001). *Usability Engineering: Scenario-Based Development of Human-Computer Interaction* (1st edition ed.). Redwood City, CA: Morgan Kaufmann.

Rosson, M. B., & Carroll, J. M. (2002). Scenario-Based Design. In J. Jacko & A. Sears (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications* (pp.1032-1050): Lawrence Erlbaum Associates.

Sarder, M. B., Ferreira, S., Rogers, J., & Liles, D. H. (2007). *A Methodology for Design Ontology Modeling.* In Proceedings of the Management of Engineering and Technology, Portland International Center for, Portland, OR, pp.1011-1018.

Sarder, M. D. B. (2006). *The development of a design ontology for products and processes.* Doctoral Dissertation, The University of Texas at Arlington, Arlington, TX.

Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R. d., Shdbolt, N., Velde, W. V. d., *et al.* (2000). *Knowledge Engineering and Management: the CommonKADS methodology*: MIT Press.

Schreiber, G., Wielinga, B., & Jansweijer, W. (1995). *The KACTUS View on the 'O' Word.* In Proceedings of the IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, pp.15.11–15.10.

Smith, B. (2003). Ontology. In L. Floridi (Ed.), *Blackwell Guide to the Philosophy of Computing and Information* (pp.155-166). Oxford, UK: Wiley-Blackwell.

Smith, B., & Welty, C. (2001). *Ontology: towards a new synthesis.* In Proceedings of the Proceedings of the international conference on Formal Ontology in Information Systems (FOIS'01), Ogunquit, Maine, USA, pp.iii-ix.

Soares, A., & Fonseca, F. (2007). Ontology-Driven Information Systems: At Development Time. *Journal of Computer, Systems and Signals, 8*(2), pp.50-59.

Solveberg, A. (1979). *Software Requirement Definition and Data Models.* In Proceedings of the Fifth International Conference on Very Large Data Bases, Rio de Janeiro, Brazil, pp.111-118.

Sowa, J. F. (2000). *Knowledge Representation: Logical Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing Co.

Staab, S., & Maedche, A. (2000). *Ontology Engineering beyond the modeling of Concepts and Relations.* In Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'2000), Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin, Germany, pp.1-6.

Staab, S., Schnurr, H. P., Studer, R., & Sure, Y. (2001). Knowledge Processes and Ontologies. *IEEE Intelligent Systems, 16*(1), pp.26–34.

Stevens, R., Goble, C. A., & Bechhofer, S. (2000). Ontology-based knowledge representation for bioinformatics. *Briefings in Bionformatics, 1*(4), pp.398-414.

Storey, V., & Purao, S. (2004). Understanding relationships: Classifying verb phrase semantics. In P. Atzeni, W. Chu, H. Lu, S. Zhou & T. Ling (Eds.), *Conceptual Modeling – ER 2004. Proceedings of the 23rd International Conference on Conceptual Modeling* (Vol. LNCS 3288, pp.336-347): Springer-Verlag Berlin Heidelberg.

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge Engineering: Principles and Methods. *IEEE Transactions on Data Knowledge and Engineering, 25*(1-2), pp.161-197.

Sugumaran, V., & Storey, V. C. (2002). Ontologies for conceptual modeling: their creation, use, and management. *Data & Knowledge Engineering, 42*(3), pp.251-271.

Sugumaran, V., & Storey, V. C. (2006). The Role of Domain Ontologies in Database Design: An Onotlogy management and Conceptual Modeling Environment. *ACM Transactions on Database Systems, 31*(3), pp.1064-1094.

Sureephong, P., Chakpitak, N., Ouzrout, Y., & Bouras, A. (2008). An Ontology-based Knowledge Management System for Industry Clusters. In X.-T. Yan, W. J. Ion & B. Eynard (Eds.), *Global Design to Gain a Competitive Edge: An Holistic and Collaborative Design Approach based on Computational Tools* (pp.333-342): Springer London.

Sutcliffe, A. (1998). Scenario-based requirements analysis. *Requirements Engineering, 3*(1), pp.48-65.

Sutcliffe, A. (2003, September 8-12 2003). *Scenario-based Requirements Engineering, mini-tutorial.* In Proceedings of the 11th IEEE Internationl Requirements Engineering Conference, Monterey Bay, CA, pp.320-329.

Sutcliffe, A. G., Maiden, N. A. M., Minocha, S., & Manuel, D. (1998). Supporting Scenario-Based Requirements Engineering. *IEEE Transaction on Software Engineering, 24*(12), pp.1071-1088.

Swartout, B., Ramesh, P., Knight, K., & Russ, T. (1997). *Toward Distributed Use of Large-Scale Ontologies.* In Proceedings of the AAAI'97 Spring Symposium on Ontological Engineering, Stanford University, CA, pp.138–148.

Takagaki, K. (1990). *A formalism for object-based information systems development.* Doctoral Dissertation, The University of British Columbia, Vancouver, Canada.

Torgerson, C. (2003). *Systematic Reviews*. London, UK: Continuum International Publishing Group.

Tun, N. N., & Tojo, S. (2006). Identity Conditions for Ontological Analysis. In J. Lang, F. Lin & J. Wang (Eds.), *Knowledge Science, Engineering and Management (KSEM 2006)* (Vol. LNAI 4092, pp.418-430): Springer-Verlag Berlin / Heidelberg.

Uschold, M. (1996). *Building Ontologies: Towards a Unified Methodology.* In Proceedings of the 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems (Expert Systems '96), Cambridge, UK.

Uschold, M. (1998). Knowledge level modelling: concepts and terminology. *The Knowledge Engineering Review, 13*(1), pp.5-29.

Uschold, M. (2003). Where are the Semantics in the Semantic Web? *AI Magazine, 24*(3), pp.25-36.

Uschold, M. (2008, Oct 31st - Nov 3rd). *Ontology-Driven Information Systems: Past, Present and Future.* In Proceedings of the 5th International Conference on Formal Ontology in Information Systems (FOIS2008), Saarbrücken, Germany, pp.3-18.

Uschold, M., & Grüninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review, 11*(2), pp.93-136.

Uschold, M., & King, M. (1995). *Towards a Methodology for Building Ontologies.* In Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, pp.6.1–6.10.

van der Vet, P. E., & Mars, N. J. I. (1998). Bottom-up construction of ontologies. *IEEE Transactions on Knowledge and Data Engineering, 10*(4), pp.513-526.

Vandana, K. (2007). *Ontology for Information Systems (O4IS) Design Methodology: Conceptualizing, designing and representing domain ontologies.* Doctoral Dissertation, The Royal Institute of Technology, Sweden.

Vrandecic, D., Pinto, S., Tempich, C., & Sure, Y. (2005). The DILIGENT knowledge processes. *Journal of Knowledge Management - Special Issue: Semantic knowledge management, 9*(5), pp.85-96.

Wand, Y. (1996). Ontology as a foundation for meta-modelling and method engineering. *Information and Software Technology, 38*, pp.281-287.

Wand, Y., Monarchi, D. E., Parsons, J., & Woo, C. (1995). Theoretical Foundations for Conceptual Modelling in Information System Development. *Decision Support Systems, 15*, pp.285-304.

Wand, Y., & Weber, R. (1989). *An Ontological Evaluation of Systems Analysis and Design Methods.* In Proceedings of the IFIP WG 8.1 Working Conference on Information Systems Concepts: An In-Depth Analysis, Namur, Belgium, pp.79-107.

Wand, Y., & Weber, R. (1990). An Ontological Model of an Information System. *IEEE Transaction on Software Engineering, 16*(11), pp.1282-1292.

Wand, Y., & Weber, R. (2002). Research Commentary: Information Systems and Conceptual Modeling: A Research Agenda. *Information Systems Research, 13*(4), pp.363-376.

Wand, Y., & Weber, R. (2004). Reflection: Ontology in Information Systems - Foreword. *Journal of Database Management, 15*(2), pp.iii-vi.

Wang, N., & Xu, X. (2000). *A method to build ontology.* In Proceedings of the The Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region, Beijing, China, pp.672-673.

Weidenhaupt, K., Pohl, K., Jarke, M., & Haumer, P. (1998). Scenarios in System Development: Current Practice. *IEEE Software, 15*(2), pp.34-45.

Welbank, M. (1987). *Knowledge Acquisition: a survey and British telecom experience.* In Proceedings of the First European Workshop on Knowledge Acquisition for Knowledge-Based Systems, Reading University: Reading, UK.

Wyssusek, B. (2004). *Ontology and Ontologies in Information Systems Analysis and Design: A Critique.* In Proceedings of the 10th Americas Conference on Information Systems, New York, NY, pp.4303-4308.

Yildiz, B., & Miksch, S. (2007, Feb 21-23). *Ontology-Driven Information Systems: Challenges and Requirements.* In Proceedings of the International Conference on Semantic Web and Digital Libraries (ICSD-2007), Bangalore, India, pp.35-44.

Yu-N, C., & Abidi, S. S. R. (2000a). *An Ontology-Mediated Scenario Composer for Knowledge Acquisition in Intelligent Systems.* In Proceedings of the TENCON 2000, Kuala Lumpur, Malaysia, pp.377-381.

Yu-N, C., & Abidi, S. S. R. (2000b). *A Scenarios Mediated Approach for Tacit Knowledge Acquisition and Crystallisation: Towards Higher Return-On-Knowledge and Experience.* In Proceedings of the Third International Conference on Practical Aspects of Knowledge Management (PAKM2000), Basel, Switzerland, pp.5-1 to 5-12.

Zlot, F., Oliveira, K. M. d., & Rocha, A. R. (2002). *Modeling Task Knowledge to Support Software Development.* In Proceedings of the SEKE'02, Ischia, Italy, pp.35-42.

Appendix A:

Summary of the Methodologies

Methodology	Knowledge Acquisition	Identify Concepts/ Relationships	Identify Tasks	Identify Temporal Relations	Identify Axioms	Ontology Levels	Mapping
OBCM/IS	Examining the application	Identifying things of interest and grouping into kinds	Change Functions and transformation of States	No information	Law statements	Bunge's Ontology	No information
CYC	Manually build the knowledge	No information	Describe some constructs for tasks, events, and processes	temporality and causality	Schemata (slots): structure constraint, activity constraint, purpose constraint	Cyc's Global Ontology (top layer), the rest of human knowledge	Examples for components of the Global Ontology.
Gruninger & Fox	Motivating Scenarios and Competency Questions	Terms extracted from the Motivating Scenarios and Competency Questions	Informal Competency Questions related to Planning and Scheduling.	Informal Competency Questions related to Temporal Projection	Informal and Formal Competency Questions. Use of predicates	Activity Ontology, Organization Ontology	No information
KACTUS	No information	List of domain-related terms	No information	No information	No information	Ontology meta-model of another ontology or a domain theory	Mapping Rules: vocabulary to vocabulary with or without semantic changes
Uschold & King	BSDM method, Scoping, Competency questions	Scoping: Brainstorming and Grouping	No information	No information	No information	Meta-Ontology	No information
Methontology	KBS knowledge elicitation techniques. Non-	Build a glossary of terms (GT) with	No information	No information	Table of axioms, Formulas and Rules	Reuse Meta-Ontology such as Cyc and	Develop an integration document

Methodology	Knowledge Acquisition	Identify Concepts/ Relationships	Identify Tasks	Identify Temporal Relations	Identify Axioms	Ontology Levels	Mapping
	structured interview, informal/formal text analysis, and structured interview, scenarios of use, competency questions	Classifications trees and verbs diagrams				Ontolingua	
SENSUS	Domain experts identify seed terms	Seed terms are linked to SENSUS, terms from the seed to the root are kept, add terms specific to the domain	No information	No information	No information	Broad coverage Ontology and domain ontology	Link domain terms to broad Sensus ontology
CAKE	Stratification process (S1-S8)	S3-WHAT-knowledge	S4-HOWTO-knolwedge, S7-WHY-knowledge	S6-WHEN-knowledge	No information	No information	No information
KIS/OBM	Domain analysis and knowledge chains	Structure Ontology: Domain knowledge dictionary, constant definition, and formula definition	No information	No information	No information	No information	No information
Chen & Chan	Relevant document study or interviews with domain experts	From the knowledge acquisition approach	No information	No information	No information	Upper Model (class, process, relations), Middle Level, and Domain Specific Level	No information
Noy & McGuinness	Determine domain and scope:	Enumerate important terms (Top-down,	No information	No information	Facets	No information	No information

Methodology	Knowledge Acquisition	Identify Concepts/ Relationships	Identify Tasks	Identify Temporal Relations	Identify Axioms	Ontology Levels	Mapping
	Competency Questions	Middle-out, Bottom-up)					
On-to-Knowledge	Usage Scenarios, Competency Questions, Interview Experts	Refinement Phase: baseline taxonomy of relevant concepts	No information	No information	From competency questions	No information	No information
Bachimont et al.	Search possible labels	Candidate Primitives, similarities and differences with parent and siblings	No information	No information	relation to relational algebra, part-whole reasoning, composition of relations, exhaustive partitions	Differential Ontology, Referential Ontology, Computational Ontology	Import and export of ontologies
Heuristic-Based	Visiting sites and browsing documents, creating use cases	Analyzing use cases	Heuristic 3.1 (Pre-requisite constraint)	Defined with the business rules. Heuristic 3.2 (temporal constraint)	Step 3: basic contraints (business rules) and Step 4: higher-level constraints	No information	No information
Bowman	Initial Modeling with problem-solving as tasks, questions and answers	Procedure 2.1: Record the objects revealed in the tasks, questions, and answers of a solution tree	Problem-solving task reduction	No information	Based on task reductions	No information	No information
Lexicon-based	Elicitation techniques (structured interviews, documents reading and questionnaire	Identify a list of terms relevant to the Universe of Discourse, Classify terms (object, subject, verb or state)	No information	No information	Derived Analysis of behavioral responses and identification of disjoint relationships	Ontology structure and LEL	No information
Jin et al.	Mind Map,	Term	No information	No information	Baseline	No information	No information

Methodology	Knowledge Acquisition	Identify Concepts/ Relationships	Identify Tasks	Identify Temporal Relations	Identify Axioms	Ontology Levels	Mapping
	Competency Questions List (CQL)	enumeration from CQL, Baseline Ontology: define classes and class hierarchy			Ontology: Define restrictions (facets) on Slots		
Diligent	Competency questions	Analysis of competency questions	No information	No information	No information	No information	No information
Gavrilova & Laird	Knowledge Elicitation techniques	Build a glossary by collecting terms	No information	No information	No information	No information	No information
Husseman & Vossen	Use case diagrams and competency questions	Relevant concepts for the specified competency questions	Requirement Analysis: Process Analysis	No information	No information	No information	No information
Brusa et al.	Motivating Scenarios and Competency Questions	List of most important terms (terms of independent existence), Concepts Classifier Tree	No information	No information	Analyzing the Concepts Classifier Trees	Domain Ontology, Formulation Ontology	No information
Kong et al.	Formatted knowledge data made by the domain experts	Select glossary from WorldNet	No information	No information	No information	No information	No information
DKAP	Acquire the raw data needed and analyze the data, Create Statement pool and other source documents, data collection techniques	Create Term Pool, identify proto properties, kinds/Kind hierarchy and relations)	Identify relevant design activities	No information	IDEF5 Elaborate Language	Site-specific ontology, practice ontology, domain ontology	No information

Methodology	Knowledge Acquisition	Identify Concepts/ Relationships	Identify Tasks	Identify Temporal Relations	Identify Axioms	Ontology Levels	Mapping
Tun & Tojo	No information	Classify sorts into groups of type, quasi-type, role and sub-role, construct sort hierarchy	No information	No information	No information	No information	No information
O4IS	Unified Semantic Procedural Pragmatic (USP^2) Design, Semantic Analysis Representations (SAR), use case scenarios	SAR: Structural Relationships, Semantic Concept View	SAR: Functional Relationships, Use case scenarios, Pragmatic and Procedural Concept View	SAR: Temporal Relationship, Procedural Concept View	SAR: Deontic and Prescriptive Relationships, Semantic and Pragmatic Concept View	Multi-tiered Domain Ontology Architecture: Upper generic, specific and application/ template ontology	Extension or restriction of the upper level ontology, specialization of the specific domain level
Ahmed et al.	Interviews and observations, literature	Elicited from engineering designers, Identify root concepts and create taxonomies	Design process and function taxonomy	No information	No information	No information	No information
Jung et al.	Use cases	Extract and listing topics from the use cases, Super- Sub- and Association type creation	No information	No information	No information	No information	No information
Sureephong et al.	Ontology Requirement Support Document (ORSD): goals, knowledge sources, scenarios, competency	ORSD: inclusion and exclusion of concepts/ relations	Task Model	No information	No information	Generic Ontology, Domain Ontology, and Task Ontology	No information

Methodology	Knowledge Acquisition	Identify Concepts/ Relationships	Identify Tasks	Identify Temporal Relations	Identify Axioms	Ontology Levels	Mapping
	questions, etc.						
OE	Domain questions (DQ), Informal and Formal Competency Questions (ICQ, FCQ)	Domain experts translate ICQs to FCQs to form final domain knowledge and keywords accurately	No information	No information	No information	No information	No information
ROC	Domain Experts identify relevant knowledge	Domain Experts create seed concepts list and define appropriate relations	No information	No information	No information	No information	No information

Vita

Andrey Soares

Andrey Soares was born in Florianopolis, Brazil. He received a Bachelor of Science in Computer Science from the University of the Valley of Itajai (UNIVALI, Brazil) in December 1998, a Graduate Diploma in Computer Science with emphasis in Computer Networks in August 2000 and a Master of Science in Computer Science in September 2001 from the Federal University of Santa Catarina (UFSC, Brazil). Andrey has completed the requirements for the Doctor of Philosophy in Information Sciences and Technology from The Pennsylvania State University in August 2009.

Prior to his academic career, Andrey worked as software engineer and system administrator at several companies in Brazil. In 1999, he started as an instructor of a community college. While at Penn State, Andrey engaged in several research and teaching activities. He spent the summer of 2008 working as a Graduate Researcher at IBM Thomas J. Watson Research Center, and received the Graduate Teaching Fellow in the Fall 2008 and the Spring 2009 from the College of IST. His areas of research interest include Ontology-Driven Information Systems, Ontology Engineering, Knowledge Representation, and Information Systems Analysis and Design.