USING ONTOLOGIES IN THE CONTEXT OF KNOWLEDGE MANAGEMENT
SYSTEMS

by

PALASH BERA
MSc B, University of British Columbia, 2003

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

In

THE FACULTY OF GRADUATE STUDIES
(Business Administration)

THE UNIVERSITY OF BRITISH COLUMBIA

August 2007

© Palash Bera, 2007
Abstract

Knowledge management systems (KMS) are information systems that combine and integrate functions for managing knowledge in organizations. Although substantial interest exists in KMS, a theoretically-based view of knowledge in the KMS context is not yet available. To clarify the notion of knowledge as managed by KMS, a conceptual framework is developed. The key concepts of this framework are derived by combining an action-based perspective with an artificial intelligence (AI) view of knowledge. The relationships among these concepts are identified, anchored to current literature, and represented graphically as conceptual models. Conceptual models are used to support the understanding of and communicating about application domains. The models contribute in proposing several theoretical and practical implications regarding KMS. To use KMS effectively, knowledge seekers need to be able to identify the knowledge required to perform their tasks. It is suggested that providing knowledge seekers with a visual representation of a formal ontology can facilitate performing knowledge identification. Formal sets of statements defining the relevant concepts and their relationships are called formal ontologies. Formal ontologies are often specified in ontological languages such as Web Ontology Language (OWL). The main requirements from such languages are that they have well-formalized syntax and that they will be computer-readable. However, not much attention has been paid to how they can be used to convey domain semantics. It is suggested that the use of philosophically-based ontological principles can help generate guidelines for developing conceptual models using OWL. Accordingly, a set of guidelines is proposed and it is demonstrated that application of such guidelines can provide clearer representation of domain phenomena such as interaction. Ontologies developed with these guidelines for modeling interaction are termed informed ontologies. From the developed conceptual models for KMS, it is identified that knowledge is intimately tied to the change of state of an entity. This change of state is facilitated by entities participating in interactions. Thus, it is proposed that the use of informed ontologies will lead to better knowledge identification than the use of uninformed ontologies. In a laboratory study, using business students as subjects, it was found that the use of informed ontologies for knowledge identification was advantageous.
# Table of Contents

Abstract.......................................................................................................................... ii  
Table of Contents........................................................................................................... iii  
List of Tables ................................................................................................................... vi  
List of Figures ................................................................................................................ vii  
Acknowledgements ....................................................................................................... ix  
Co-Authorship Statement ............................................................................................ x  
Chapter 1 Introduction ................................................................................................. 1  
  1.1 Background .......................................................................................................... 1  
  1.2 Research questions and objectives ..................................................................... 3  
  1.3 Approach and methodology ............................................................................... 3  
  1.4 Overview of the thesis ....................................................................................... 4  
Chapter 2 Conceptual Models for Knowledge Management Systems .................. 9  
  2.1 Introduction ......................................................................................................... 9  
  2.2 Foundations for Conceptual Models of KMS ..................................................... 10  
    2.2.1 The fundamental view of knowledge ............................................................ 10  
    2.2.2 Knowledge-related constructs ...................................................................... 12  
  2.3 Modeling Knowledge in the KMS Context ....................................................... 12  
    2.3.1 Knowledge from an agent’s perspective ....................................................... 12  
    2.3.2 Knowledge as used by an agent .................................................................. 20  
    2.3.3 Knowledge as managed by KMS ................................................................. 22  
  2.4 Uses and Implications of the Models ................................................................. 26  
    2.4.1 An Example of Using the Constructs ............................................................ 27  
    2.4.2 Knowledge identification, tagging, and queries ......................................... 27  
    2.4.3 On the Distinction between Knowledge and Information ........................... 28  
    2.4.4 Differentiating the roles of IS and KMS ...................................................... 30  
    2.4.5 Guidelines for KMS Development ............................................................... 31  
  2.5 Comparing the Models to Related Work ........................................................... 35  
    2.5.1 Comparing to the Learning-Oriented KMS Model ...................................... 35  
    2.5.2 Analyzing other Definitions of Knowledge ............................................... 36  
    2.5.3 Comparing to Knowledge views ................................................................. 37  
  2.6 Conclusion and Future Work ............................................................................. 39
4.5.3 Experimental setting ................................................................. 102
4.6 Results and Discussion .............................................................. 106
  4.6.1 The effect of language in the results of the study ...................... 110
4.7 Conclusion ................................................................................. 111
Chapter 5 Conclusion .................................................................... 126
  5.1 Review ...................................................................................... 126
  5.2 Contributions .......................................................................... 128
  5.3 Future Work ............................................................................ 130
Appendix A : Organizational Knowledge Definitions ....................... 133
Appendix B : Brief Description of OWL ........................................... 136
Appendix C : Specific Rules to Model Interactions ............................ 138
Appendix D : Training materials used in the experiment in Chapter 4 ...... 139
Appendix E : Domain descriptions and modifications made in the ontologies for the experiment in Chapter 4 ........................................ 143
Appendix F : Ontologies used in the experiment described in Chapter 4 .... 145
Appendix G : Mapping Concepts between Ontologies .......................... 151
Appendix H : Test Materials of the experiment in Chapter 4 ............... 156
Appendix I : Ethical Review Certificate obtained for the empirical study in chapter 4 .. 160
Appendix J : Coding Manual For the study in Chapter 4 ....................... 161
Appendix K : Analysis of the terms used in the ontologies ................... 167
List of Tables

Table 1: Statements about relationships of knowledge-related concepts ........................................... 17
Table 2: Statements underlying a conceptual model of KMS ............................................................... 25
Table 3: Examples of the knowledge related constructs ........................................................................ 27
Table 4: Knowledge views (Alavi and Leidner, 2001, p. 110) .............................................................. 38
Table 5: Concepts of OWL (adapted from (McGuinness et al., 2004)) ............................................... 50
Table 6: Concepts of Bunge’s ontology (Wand and Weber, 1993) ....................................................... 56
Table 7: Ontological approaches on Entity, Class, and Property ......................................................... 57
Table 8: Guidelines derived from ontological theory for modeling interactions in OWL ............... 95
Table 9: One way ANOVA for differences Between Groups ............................................................. 104
Table 10: ANCOVA analysis for knowledge identification tasks ....................................................... 105
Table 11: Descriptive Statistics for the variables ................................................................................. 106
Table 12: Instrument validity and reliability ....................................................................................... 107
Table 13: Confidence Rating of Coders ............................................................................................... 108
Table 14: MANOVA analysis for individual items of knowledge identification task .................... 109
Table 15: ANOVA analysis for hypotheses testing .............................................................................. 109
Table 16: ANCOVA analysis for knowledge identification tasks ..................................................... 109
Table 17: Analysis of definitions of knowledge in KM literature ..................................................... 134
Table 18: Summary of knowledge related concepts that appear in knowledge definitions (in Table 17) ................................................................................................................................. 135
Table 19: Summary of other concepts that appear in knowledge definitions (in Table 17) ............ 135
Table 20: Establishing information equivalency between the travel ontologies (from uninformed to informed) .................................................................................................................... 152
Table 21: Establishing information equivalency between the travel ontologies (from informed to uninformed) ........................................................................................................................ 153
Table 22: Establishing information equivalency between the auction ontologies (from uninformed to informed) .................................................................................................................... 154
Table 23: Establishing information equivalency between the auction ontologies (from informed to uninformed) ........................................................................................................................ 154
Table 24: Establishing information equivalency between the auction ontologies (from informed to uninformed) ........................................................................................................................ 155
Table 25: Analysis of the terms used in the ontologies ...................................................................... 169
List of Figures

Figure 1: Connection among the three parts of the thesis ................................................. 6
Figure 2 : Statement 1: There are two types of entities: agents and non-agents ............... 13
Figure 3. Statement 2: An entity has an environment comprising other entities ............. 14
Figure 4. Statement 3: An entity has states ..................................................................... 14
Figure 5. Statement 4: An agent has goal ........................................................................ 14
Figure 6. Statement 5: Current states of affairs and goals are states .............................. 15
Figure 7. Statement 6: An agent is aware of the current state of affairs ........................ 16
Figure 8. Statement 7: An agent can perform actions ..................................................... 16
Figure 9. Statement 8: Actions can change state of entities ........................................... 16
Figure 10. Model of knowledge- an agent’s environment .............................................. 17
Figure 11. Quaternary relationship among agent, state, goal, and actions ................... 19
Figure 12. Model of knowledge- an agent’s perspective ................................................. 20
Figure 13. Model of knowledge as used by an agent ...................................................... 21
Figure 14. Statement 1: Knowledge has representations .................................................. 23
Figure 15. Statement 2: Knowledge representation can be explicit or implicit ............... 23
Figure 16. Statement 3: KMS stores explicit representation of knowledge .................... 23
Figure 17. Statement 4: KMS points at representation of knowledge .............................. 24
Figure 18. Statement 5: Implicit representation of knowledge are contained in containers 24
Figure 19. Statement 6: Containers can be agent and organizational transformers ......... 25
Figure 20. Model of KMS .............................................................................................. 26
Figure 21. Ontology describing knowledge related concepts ......................................... 34
Figure 22. Ontology representing knowledge related concepts with instances (IO refers to instances) ......................................................................................................................... 35
Figure 23: OWL property associated through classes (“IO” means “Instance Of”) .......... 64
Figure 24: Original Wine Ontology (“IO” means instance of”; “U” means “union”) ...... 65
Figure 25: Revised Wine Ontology .................................................................................. 66
Figure 26: WineDescriptor hierarchy ............................................................................. 67
Figure 27: Example of OWL object property ................................................................ 69
Figure 28: Unguided OWL representation of the customer-bank domain .................... 70
Figure 29: Class hierarchy of the bank-customer domain .............................................. 71
Figure 30: OWL representation of the bank customer domain showing one interaction 73
Acknowledgements

At the outset, I'm indebted to Dr. Yair Wand, my supervisor, for his support, patience, and advice. His timely advice helped me to remain focused on the research. He showed me the way on how to be a good academician (i.e. in the role of a researcher, a teacher, a reviewer, and a guide).

Next, I thank my committee members (Dr. Joseph Tennis and Dr. Andrew Burton-Jones) who kindly agreed to be in my committee. Their suggestions and advice were very useful. In particular, the committee members helped me to understand the research problem from different perspectives. I thank Dr. Andrew Burton-Jones for providing suggestions on the design of the empirical study. I am also very grateful to Anna Kräsonperova for fruitful discussions, advice, and feedback on my work.

Last but not least, this journey would not have been complete without the support of my family (wife - Veena and daughter - Disha). Veena has been very understanding and caring during my life as a student. Little Disha, an avid fan of her father, is a constant source of inspiration.
The following describes the contributions of the co-authors in developing the chapters. Dr. Yair Wand helped me to identify the research themes (chapter topics) and suggested the design of the research program. In Chapter 3 (section 3.4), Anna Krasnoperova developed modeling rule 1 and guidelines 3 and 4. Anna Krasnoperova also developed guidelines 5, 6, 8 and 9 (in section 3.5). In chapter 4, Dr. Andrew Burton Jones helped to design the empirical study, in particular by suggesting the use of training to operationalize the knowledge identification task and the use of feedback to operationalize the perceived variables.
Chapter 1
Introduction

1.1 Background

The importance of managing knowledge in organizations (commonly termed *knowledge management*) is widely recognized. Gartner (2005a) reports that by 2005, most leading organizations have implemented knowledge management (KM) practices to support at least one critical business process, and many have implemented enterprise-wide KM programs. To support these programs, organizations often deploy *knowledge management systems* (KMS) which are information systems (IS) applied to the management of organizational knowledge (Alavi and Leidner, 2001). However, the term “knowledge” in the context of KMS lacks clarity. Confusion exists on what KMS really are. For example, Maier (2002, p. 42) mentions “more traditional software like document management systems, data warehouses, and analysis tools are marketed increasingly as knowledge management systems” and Moffet et al. (2003, p. 8) mention that “confusion still exists over it’s implications for KM. One of the main reasons for this has been the re-packaging of software applications under the KM label.”

To perform a task, question of particular interest is how users identify the knowledge they require to perform their tasks. This task is named *knowledge identification*. To perform knowledge identification, users need to be familiarized with the domain concepts. This familiarization can be especially useful if users have inadequate domain understanding. Accordingly, the use of *ontologies* might help to identify the knowledge required to perform a task. An ontology is a set of concepts and relationships among these concepts intended to describe a particular domain (Gomez-Perez et al., 2004). A formal ontology (often termed IS ontologies) expresses the concepts and relationships in a machine-readable form. IS ontologies have been introduced in IS because of their promise to improve *communication, sharing*, and *reuse* of information in the Web context (Uschold and Gruninger, 1996). Languages such as Web Ontology Language (OWL) are often used for representing formal ontologies.
Industry reports such as Gartner (2005b) mention that more than 80% of the information used by organizations is unstructured (such as Microsoft word documents) and predicts that by 2007, 60% of information access implementations will combine search, ontology and information visualization technologies. These observations indicate that the study of KMS and in particular the use of ontologies in KMS is important and relevant.

There are two problems related to the use of formal ontologies in knowledge identification. First, ontology description languages are designed for software applications rather than to support domain understanding. Therefore, they might not be able to convey domain semantics properly, and might lack the capacity to reflect some important domain phenomena (Bera and Wand, 2004). Second, knowledge seekers cannot be expected to understand formalized ontologies. As well, no rules exist on how to develop and present formal ontologies (Gomez-Perez et al., 2004).

The thesis addresses the following issues. First, how to develop a clear definition of knowledge in order to understand how knowledge is managed by KMS? Second, how to improve ontological languages so that they can convey domain semantics clearly? Third, what role does ontology play in identifying the required knowledge in the KMS context? How to provide any empirical evidence to support this role?

Each of these issues is addressed in this thesis as three chapters. In chapter 2, the objective is to clarify the definition of knowledge in the context of KMS. Chapter 3 discusses how OWL can be improved to represent domain facts. Models used to support the understanding of and communicating about application domains in the development of IS are called conceptual models (Mylopoulos, 1992). Such models are expressed in conceptual modeling languages. Chapter 3 addresses how OWL can be used as a conceptual modeling language to represent domain facts accurately. In chapter 4 an additional role of ontology in the context of KMS is identified. Ontologies are suggested for the purpose of identifying knowledge required for a task. To use KMS effectively, knowledge seekers need to be able to identify the knowledge required to perform their tasks. Hence, a question of particular interest is how users identify the knowledge they require to perform their tasks. This task is named knowledge identification. To perform knowledge identification, users need to be familiarized with the domain concepts. This
familiarization can be especially useful if users have inadequate domain understanding. It is suggested that providing knowledge seekers with a visual representation of a formal ontology can facilitate performing knowledge identification. This suggestion is motivated by the fact that the same ontology can be used to organize and access knowledge. Chapter 4 provides empirical evidence to support the proposition that visual ontologies can be useful to facilitate performing knowledge identification. In the next section, these issues are linked to the objectives and research questions.

1.2 Research questions and objectives

The overall theme of the thesis is: “How to use ontologies effectively for using Knowledge Management Systems?” This question can be further divided into three objectives:

1. formalize the role of knowledge in the context of Knowledge Management Systems
2. suggest to use formal ontology languages (such as OWL) in the context of Knowledge Management Systems
3. test the use of visual formal ontologies in the context of knowledge search in Knowledge Management Systems

The first objective is to clarify the concept of knowledge in the context of KMS. In particular, define terms such as knowledge identification and knowledge tagging, differentiate KMS from other related IS and help suggest guidelines for designing KMS. The second objective is to suggest how to use OWL as a conceptual modeling language. In particular, the focus is to develop specific guidelines for creating conceptual models in OWL and, via examples, demonstrate how the application of the guidelines can provide clearer representation of domain phenomena. The third objective is to empirically test the usefulness of models developed in OWL to assist performing knowledge identification tasks. This objective is realized by developing propositions and testable hypotheses and then testing these hypotheses in a controlled laboratory experiment. In the next section, it is discussed how these objectives were realized.

1.3 Approach and methodology

To develop the conceptual models for KMS an ontological approach was taken. In this approach, it is believed that knowledge can be expressed in terms of a set of concepts and their
relationships. Following this approach, the methodology adopted was as follows. A set of concepts: agent, action, state and goal were derived by combining action-based perspectives with artificial intelligence (AI) view of knowledge. The relationships among these concepts were identified and anchored to current literature about knowledge. The concepts and their relationships formed the basis for three conceptual models describing knowledge from an individual agent’s perspective, knowledge as used by an agent, and the role of KMS in providing access to knowledge.

To use OWL as a conceptual modeling language, the approach was to develop guidelines using philosophy based ontologies. In philosophy, ontology is the branch that deals with the order and structure of reality in the broadest sense possible (Angeles, 1981). Hence, the ontological theories can provide guidance on how to represent domain phenomena. The application of the guidelines can provide clear representation of domain phenomena and thus create meaningful models in OWL. Following this approach, the methodology adopted here was to use a particular philosophical ontology- Bunge’s ontology (1977; 1979) to prescribe guidelines for using OWL as a conceptual modeling language.

To test the effectiveness of the models developed in OWL an empirical study was conducted. It is suggested that knowledge identification can be facilitated by providing users with information about the relevant concepts underlying a domain. For this purpose, models developed in OWL can be used. Two sets of models were developed in OWL. One set of models was developed according to guidelines reflecting a philosophically based ontology (Bunge’s ontology) and the other set of the models was developed without the use of any guidelines. Students from a business school were recruited as subjects in this study. The purpose of the study was to test whether the use of conceptual model developed with philosophically based ontological guidelines would lead to better knowledge identification than the use of conceptual models developed without such guidelines.

1.4 Overview of the thesis

This section presents the overview of the thesis and explains how the three parts: (1) conceptual models for KMS, (2) using OWL as a conceptual modeling language and (3) ontological support for knowledge identification: a theory and an empirical study, of the thesis are connected.
In the first part of the thesis, the key concepts to describe the domain of knowledge and KMS are identified. The models are presented graphically and related to current knowledge management (KM) literature. These models provide the definition of knowledge in the context of KMS. Based on these definitions, knowledge related terms: knowledge identification, knowledge tagging and knowledge query are defined. Practical guidelines to develop KMS are derived from these models. The conceptual models of knowledge guided to differentiate knowledge from information. This differentiation was used to suggest the difference between IS and KMS.

In the second part of the thesis it is proposed that to create meaningful conceptual models appropriate modeling guidelines can be used. Ontology description languages such as OWL (Web Ontology Language) have been proposed for creating formal representations of terminologies underlying heterogeneous information sources. The main requirements from such languages are that they have well-formalized syntax and that they will be computer-readable. However, not much attention has been paid to how they can be used to convey domain semantics. It is suggested that the use of philosophically-based ontological principles can help generate guidelines for developing conceptual models using OWL. The second part of the thesis presents specific guidelines for creating conceptual models in OWL and, via examples, demonstrates how the application of the guidelines can provide clearer representation of domain phenomena.

In the third part of the thesis it is proposed that knowledge identification can be facilitated by providing users with information about the relevant concepts underlying the problem domain. Such information can be provided in the form of formal ontologies developed in OWL. Formal set of statements defining concepts and their relationships and termed formal ontologies. Formal ontologies are designed for software applications rather than to support domain understanding. Therefore, they might not be able to convey domain information properly. It is suggested that the usefulness of formal ontologies can be enhanced by constructing them according to guidelines reflecting a philosophically based ontology. Formal ontologies developed with such guidelines are termed as informed ontologies. A laboratory study is described that compares the use of visual forms of informed and uninformed ontologies. The purpose was to test whether the use of visual representations of informed ontologies would lead
to better knowledge identification than the use of uninformed ontologies. The results indicated that the use of informed ontologies was advantageous in terms of helping subjects to identify knowledge needed to perform a task. Furthermore, subjects perceived informed ontologies to be more understandable than the uninformed ones.

Figure 1 describes how the above-mentioned three parts are connected. Using Entity relationship (ER) notations, three statements are represented in the Figure. These statements are as follows:

1. *Conceptual models for KMS* provide definition of *knowledge identification*
2. *Knowledge identification* can be performed using ontologies in *OWL*
3. *OWL* can be used as a *conceptual modeling language*

The concepts (in italicized) from these statements are represented in rectangles. Arrows indicate the directions of the relationships among these concepts.

![Figure 1: Connection among the three parts of the thesis](image)

Statement 1 relates to the part described in Chapter 2. This chapter provides definition of knowledge in the context of KMS. Further, the chapter provides the definition of the term knowledge identification. Statement 3 relates to Chapter 3 on how OWL can be used as a conceptual modeling
language. Specifically, from this chapter it is obtained that ontologies in OWL can be developed using philosophically based ontological theories. Such theories provide guidance on developing modeling guidelines that allow construction of ontologies in OWL. To test the usefulness of the conceptual models in assisting knowledge identification, statement 2 is proposed. This statement relates to Chapter 4. In this chapter an empirical study is described that compares two sets of conceptual models in their ability to facilitate knowledge identification. One of these sets of conceptual models are developed using modeling guidelines discussed in Chapter 3 while the other set of conceptual models are developed without any guidelines.
References


Chapter 2
Conceptual Models for Knowledge Management Systems

2.1 Introduction

Organizations are willing to invest in the acquisition and development of knowledge management systems (KMS) because they recognize the importance of managing knowledge. However, surprisingly, it appears that confusion still exists on what KMS really are. For example, Maier (2002, p. 42) claims: “more traditional software like document management systems, data warehouses, and analysis tools are marketed increasingly as knowledge management systems” and Moffet et al. (2003, p. 8) mention that “[a]lthough the technological arena has received much publicity in recent years, confusion still exists over it’s implications for KM. One of the main reasons for this has been the re-packaging of software applications under the KM label.” Vaast et al. (2006, p. 316) reflect this issue by stating that “as new generations of KMS systems have come to the market, replacing “old” expert systems with recommender systems, for instance, the very definition of what the “knowledge” is that has to be managed with KMS has shifted. These shifts need to be better understood.”

Lack of clarity is also observed in the use of the term “knowledge” in the KM context. Brown and Duguid (2001, p. 198) claim “The [KM] literature still presents sharply contrasting and at times even contradictory views of knowledge.” Bryant (2000) mentions that many definitions of KM could easily be replaced with similar ones related to information or to general resource management. The diversity of definitions of knowledge is apparent when considering some of the definitions that appeared in KM literature (Table 17, Appendix A). On a more general note, Spender (2003) claims that the theoretical foundations of knowledge in the context of organizations are still weak. Finally, Styhre (2003, p. 38) indicates the importance of defining the term “knowledge” in the KM context: “knowledge management theory should

---

benefit from emphasizing "knowledge" rather than "management"; only then understanding knowledge management will be possible."

The above discussion suggests that the study of KMS requires a clear definition of knowledge in order to understand what should be managed by KMS and how it should be managed. In this paper we propose a definition of knowledge in the KMS context. We base the definition on the observation that in organizational context knowledge is possessed, created, and used by agents who perform tasks. The paper considers three aspects of knowledge. First, knowledge as possessed and applied by an agent, second, knowledge as an object used by agents, third knowledge as managed by KMS. The definition of knowledge enables us to clarify how a KMS differs from a typical information system and to suggest some specific guidelines for KMS development.

Our analysis involves first, the identification of a set of constructs necessary to define knowledge from an agent's perspective and second understanding the relationships that exist among these constructs. The concepts and their relationships are used to create three conceptual models which we present in visual forms. By a conceptual model we mean a description of some aspects of the physical or social reality (Mylopoulos, 1992), as opposed to models intended to convey the design aspects of an information system. The models describe the roles of knowledge as possessed by an agent, as an object created by and used by agents and as a represented concept managed by KMS.

In the following, Section 2.2 describes the foundation for the conceptual models; Section 2.3 presents the proposed models; Section 2.4 illustrates the models with examples and suggests some uses of the models in the KMS design context; Section 2.5 relates the models to the KM literature. Section 2.6 summarizes the work and indicates directions for further research.

2.2 Foundations for Conceptual Models of KMS

2.2.1 The fundamental view of knowledge

Our purpose is to develop conceptual models to formalize KMS concepts. Since a conceptual model is constructed in order to represent a domain, it is important that the concepts needed to
describe the domain be well-defined (Wand et al., 1999). Therefore, we first suggest a set of knowledge-related concepts.

The main premise of our analysis is that in an organizational context, knowledge is created, possessed, maintained, and used by agents. The agents work in the social environment of the organization and perform actions to achieve some goals. Hence, we define knowledge from an agent's perspective. We therefore use a combination of two views: (1) knowledge as applied in a social setting, and (2) knowledge as related to actions taken by agents in the organization. We now turn to the literature that addresses these views.

Social theories of knowledge view knowledge as socially constructed and influenced by society (Dant, 1991; Stahl, 2000). Morrow and Brown (1994) propose two views of knowledge in a social context. First, knowledge is present in the society into which the individual is socialized. Second, knowledge is a resource shared by the members of the society. The pragmatism approach (Blosch, 2001) anchors knowledge in action and relates actions to successful manipulation of the real world.

The KM literature supports the view that only agents (not non-agents) create, possess or use knowledge. Elst et al. (2003) mention that agents can be seen as a natural metaphor to model KM environments which can be conceived as consisting of a number of interacting entities. Nabeth et al. (2003) mention that a KM environment comprises a set of knowledge resources, mechanisms, and agents. In this environment, the agents access the resources, participate in the creation of new knowledge resources, interact, and exchange knowledge. Blosch (2001) claims that it is only people who have knowledge and not information systems. Related to individuals, May and Taylor (2003) mention that knowledge is developed whenever an individual applies his or her concepts and interpretations, and relates conceptually to the set of commonly observed phenomena in the world. Hassell (2007) mentions that there is no knowledge outside of experience and knowledge is therefore always embodied in persons. Similarly, Hall and Paradice (2004) mention that only individuals determine the desired state, interpret environmental variables, make temporal considerations, select a solution, and determine checkpoints at which to check for progress toward the desired state. In summary, the literature supports the assumptions that only agents choose actions and are able to change states.
2.2.2 Knowledge-related constructs

We have adapted a view of knowledge as related to agents who work in an environment, performing actions to attain goals. Since we seek to define knowledge in the context of KMS, we now consider representation of knowledge as addressed in the AI literature. In particular, we apply Newell’s (1982) theory of knowledge as it employs the notions of agent and action. This view of knowledge is widely recognized in the AI community (Musen, 2004). Newell distinguishes knowledge from its representation and defines two levels: the knowledge level that deals with the nature of knowledge, and the symbol level that deals with the representation of knowledge. Each level comprises components, laws of composition that enable components to be assembled into systems, and laws of behavior (Newell, 1982). In Newell’s words: “The gross anatomical description of a knowledge-level system is simply (and only) that the agent has as parts, bodies of knowledge, goals and action” (Newell, 1982, p. 99). Thus, the knowledge level attributes to an intelligent system (an agent): knowledge, goals, and actions. The agent has a body (a physical system) with which it can act on the environment and processes its knowledge to determine what actions to take in order to attain its goals.

The models of knowledge we develop address two questions. First, what is knowledge from an agent’s perspective and, second how to represent knowledge as an object of use. The first question refers to Newell’s knowledge level and the second refers to his symbol level.

2.3 Modeling Knowledge in the KMS Context

In this section we develop three models. First, knowledge as possessed by an agent, second, knowledge as used by agents, and third knowledge as represented in KMS.

2.3.1 Knowledge from an agent’s perspective

An agent can be defined as any entity which is able to act on the environment it inhabits (Castelfranchi, 1998; OMG, 2000). We use the term “agent” to refer to an individual or a group working cohesively, acting in a given role in the organization where it possesses, uses or seeks knowledge to perform tasks. Wooldridge and Jennings ((1995) identify the following properties of agents. (1) Being autonomous namely - able to operate independently and take independent decisions. (2) Being proactive namely -exhibiting goal-directed behavior. (3) Being reactive –
namely responding to changes in the environment often involving choosing alternate course of actions if a planned action does not result in achieving a goal. (4) Having social ability to interact with other agents. Note, we do not address aspects of agents not related to knowledge (such as proactive or reactive).

We now propose a set of statements to form the basis for the proposed model of knowledge from an agent’s perspective. We depict each statement in a diagram using Entity-Relationship (ER) notation.

We begin with the following premise: *users of a KMS operate in an organizational environment comprising entities.*

We distinguish between agents and non-agents (or resources). Wooldridge (2000) defines an agent as an entity that acts upon the environment it inhabits. Huhns and Singh (1998) indicate that agents are able to act in their environment and Castelfranchi (1998) states that agents can cause changes in the environment. Luck and Inverno (2001) view agents as resources with a non-empty set of goals. In contrast, Rudowsky (2004) mentions that a resource may exhibit autonomy over its state but not over its behavior. Thus, we consider agents and resources as sub-types of a more generic type – entity (Figure 2).

![Figure 2](image_url)

**Figure 2.** Statement 1: There are two types of entities: agents and non-agents

The environment of an agent consists of all entities external to the agent, including possibly other agents (Brafman and Tennenholtz, 1997). This environment contains a collection of objects that the agent is able to at least partially perceive and manipulate (Logie et al., 2003). Accordingly, we suggest that the environment of an agent comprises entities (Figure 3).
Figure 3. Statement 2: An entity has an environment comprising other entities

Agents and their environment have states (Brafman and Tennenholtz, 1997) (Figure 4). Brafmen et al. (1998) discuss the concept of configuration space to define states. Configuration space is a set of possible configurations an entity (such as agents or objects) of interest can have.

Figure 4. Statement 3: An entity has states

An agent’s behavior is aimed at producing some result, thus we refer to a goal-oriented agent (Conte and Castelfranchi, 1995) (Figure 5). Since a goal-oriented agent may have more than one goal in a given situation, it must have some form of choice or decision (Castelfranchi, 1998). Accordingly, an agent has to decide which goals it will try to achieve in a given situation (Meyer et al., 1999).

Figure 5. Statement 4: An agent has goal
We distinguish between the current state of affairs and goals. Goals are a subset of the agent’s and its environment possible states and reflect the desires the agent intends to fulfill (Logie et al., 2003; Luck and Inverno, 2001). Castelfranchi (1998, p. 161) defines goal as a “mental representation of a world state or process that is candidate for controlling and guiding action” and indicates that goals are used for “determining the action search and selection” and “qualifying its success or failure.” In this vein we view goals as states² (Figure 6).

![Figure 6. Statement 5: Current states of affairs and goals are states](image)

To take actions, an agent must have information about its current state and that of its environment (Brafman and Tennenholtz, 1997). We thus propose that “an agent is aware of the current state of affairs” (Figure 7). Given the environment comprises entities that have states, the last proposition can be phrased: “an agent and its environment have states, which are observable by the agent.” Weinberger (1998) mentions that agents must have the cognitive capacity to perceive information about the given situation. The information is relevant as an agent bases its actions on it (Castelfranchi, 1998). It is possible that this information might be incomplete (Huang et al., 1996). In this context, Son and Baral (2001) make a distinction between state of the world and states of the agent’s knowledge about the world. The former represents any state of the world, while the latter is about that part of the state of the world the agent can manipulate. We thus posit that agents and objects with which the agent cannot interact or manipulate do not belong to the agent’s environment. The state we refer to therefore reflects only the relevant entities. This can be viewed as part of defining the context in which the agent operates.

² Please note that the arrows (indicating “IS A” relation) in Figure 6 (and in Figures 10, 12, 13, 15, 19, and 20) do
An agent possesses a "repertoire of possible actions it can perform to modify the current state (of itself or of its environment) (Wooldridge, 2000) (Figure 8). Actions are always attributed to agents (Weinberger, 1998). Every action must change something, otherwise it does not matter. Henceforth we only refer to actions that can change the (observable) states of the agent or its environment.

The actions of an agent can change its own state or the state of its environment. Wooldridge (2000) mentions that a set of actions is required to change the state of an agent. Castelfranchi (1998, p. 160) indicates that an agent can “produce some causal effect and some change in its environment” and that “action is aimed at producing effect on the environment.” Since we view the environment as comprising entities, we include the statement that agent’s actions can change states of entities (Figure 9).

not indicate exhaustiveness, i.e. states other than goals and current state of affairs are possible (e.g. future states).
We summarize the above observations in terms of eight statements linking the five concepts of agents, environment, states, goals and actions (Table 1).

<table>
<thead>
<tr>
<th>Statements about relationships of knowledge-related concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>• There are two types of entities: agents and non-agents (resources)</td>
</tr>
<tr>
<td>• An entity has an environment comprising other entities</td>
</tr>
<tr>
<td>• An entity has states</td>
</tr>
<tr>
<td>• Agents have goals</td>
</tr>
<tr>
<td>• The current state of affairs and goals are states</td>
</tr>
<tr>
<td>• An agent is aware of the current state of affairs</td>
</tr>
<tr>
<td>• An agent can perform actions</td>
</tr>
<tr>
<td>• Actions can change states of entities (the agent or other entities)</td>
</tr>
</tbody>
</table>

Combined together, these statements form the basis for the proposed conceptual model of knowledge as related to agents. We depict these relationships in a diagram using the ER notation where each of the five concepts is represented as an entity type and each statement as a relationship (Figure 10).

Figure 10. Model of knowledge- an agent’s environment
Having identified the main concepts needed to define knowledge and their relationships, we now turn to the definition of knowledge in the agent context. Newell’s view of knowledge combines several concepts under the statement: “the agent has as parts bodies of knowledge, goals and actions. They are all connected together in that they enter into determination of what actions to take” (Newell, 1982, p. 99). Furthermore, he proposes that “knowledge is a competence-like notion, being a potential for generating action” (1982, p. 100).

The outcomes of an action depend on the state prior to the action. Brafman and Tennenholtz (1997, p. 225) mention that “the effects of the agent’s actions are a function of its state and the environment state.” (Castelfranchi, 1998, p. 160) indicates that: “[a]n agent receives and exploits relevant information from and about the world…” and “[o]ur agent bases its action on it.” Finally, Brafman and Tennenholtz (1997, p. 238) also note that “that we must view the agent as choosing between protocols [a series of actions] rather than single actions, since a single action produces a single next state while a protocol produces a sequence of states.”

Based on the above, we define knowledge in the following statement:

*Given the states of the agent and the environment, knowledge is the ability of the agent to select actions (from those available to the agent) so as to change the current state of affairs to a goal state.*

We depict this statement as a quaternary relationship connecting the agent, state, goal, and actions (Figure 11). We note that the state in this statement is that of the environment, including the agent.
We make several notes regarding our proposed definition of knowledge. First, Newell mentions that “selection of action does not mean it is executed, it just becomes a candidate for execution” (Newell, 1982, p. 102). Thus we suggest that if the selected actions are actually performed by the agent then they provide a manifestation of knowledge. Second, the definition does not imply the selection of actions is “right” in terms of actually being able to bring about the desired outcomes. This has to do with veracity of the knowledge. All the agent can do is to act according to beliefs about the outcomes of actions. The agent-related literature views beliefs as an important aspect of the definition of an agent. For example, Bratman (1987, p. 15) states “[a] belief is some aspect of the agent’s knowledge or information about the environment, self or other agents.” Third, whether the agent should be included as part of the definition or not, depends on the context. The agent should be shown explicitly if the types of actions available depend on the agent (or rather on its role).

We also note that none of the specific statements that we presented here is new. However, the combination of the individual statements in a meaningful way, and their use for explicitly defining knowledge in terms of several agent-related concepts are, to the best of our knowledge, new.
We complete the discussion of the agent-based definition of knowledge by combining the two diagrams — one describing the agent's environment (Figure 10) and one depicting the definition of knowledge (Figure 11). For simplicity, we have omitted aspects of the environment in the combined diagram. The combined model is shown in Figure 12.

![Diagram](image)

Figure 12. Model of knowledge- an agent's perspective

**2.3.2 Knowledge as used by an agent**

Above we defined knowledge as possessed by an agent. Recall, our goal is to define knowledge in the KMS context. The purpose of KMS is to enable agents to use knowledge created by other agents. Hence we now turn to modeling how agents use knowledge that is already available. Newell (1982, p. 99) mentions that “knowledge can be defined independent of the symbol level.” In this context, once given, knowledge can be considered as available somewhere (and its representation might be stored in a technological system). The discussion above points out that an agent uses knowledge to determine actions necessary to attain goals. Thus we suggest a relationship “use” between an agent and knowledge (viewed as an entity). This relationship links
together knowledge, agent, current state of affairs, goals (desired states) and actions under the statement:

An agent uses knowledge to choose actions to change the current state into a goal state.

The "use" relationship is the centre of a proposed model of knowledge as an object of use (Figure 13). This model depicts an agent taking actions to bring about a specific goal while being aware of the current state of affairs. It is closely related to the previous model, with one notable difference. Knowledge⁴ is now viewed as an entity (in the diagram). This is because we are interested in exploring KMS, the purpose of which is to support management of knowledge. Note, the diagram does not indicate who the source of the knowledge is. It can be the agent or other agents.

Figure 13. Model of knowledge as used by an agent

⁴ Please note that the term “knowledge” in this Figure refers to the knowledge as “an object of use” and not the concept of knowledge as used in Figures 11 and 12
To conclude this section we make two notes regarding the notion of agent in the models. First, agents can exist independently or work in groups. Therefore knowledge can be possessed by individual agents or by a collection of agents. Using this distinction between individuals and groups, Alavi and Leidner (2001) use the terms individual knowledge (for an individual agent) and social knowledge (for a group of agents). However, this distinction would not change our definition of knowledge. Second, our definition does not distinguish between human and software agents. Rather, it requires that an agent can choose and take actions based on some awareness of current states and a notion of goal.

2.3.3 Knowledge as managed by KMS

Above we have suggested a definition of knowledge as possessed and used by agents in the organization. We now develop a model of knowledge as managed by KMS.

The KM literature views the role of KMS as providing for or supporting the creation, gathering, organizing, and disseminating organization’s knowledge (Alavi and Leidner, 2001; Wakefield, 2005). We consider KMS as systems for storing the explicit representation of knowledge, or pointing at knowledge as a resource embedded in some implicit form in the organization.

Newell mentions (1982, p. 99) that “knowledge can be defined independent of the symbol level but can also be reduced to symbol level.” This statement conveys the idea that (some) knowledge can be symbolically represented. Vaast et al. (2006) observe that representations of knowledge in early KMS were explicit and visible (e.g., rules and keywords), whereas knowledge representations in latter systems increasingly tend to be implicit and invisible. Maier (2002) refers to the implicit knowledge as informal knowledge (such as ideas and FAQ’s) and mentions that KMS manages informal knowledge. Based on the above, we consider knowledge an abstract notion, that can be represented in explicit (formal) or implicit (non-formal) symbolic forms. Thus representation refers to both explicit and tacit knowledge (Nonaka and Takeuchi, 1995).

We formalize the above in the propositions that knowledge (as managed by a KMS) has representation (Figure 14) and that it has two forms – explicit and implicit (Figure 15).
The KM literature classifies KMS into repository and network (Alavi, 2000) or repository and dictionary types (Gallupe, 2001), depending if their role is to store knowledge or to point at where knowledge is located.

The repository model focuses on the storage, and retrieval aspects of knowledge (Wakefield, 2005) and views knowledge as an object that can be collected, stored, organized, and disseminated. Typically knowledge repositories store documents with knowledge embedded in them (Kwan and Balasubramanian, 2003). Thus a KMS can store explicit representations of knowledge (Figure 16).

Figure 14. Statement 1: Knowledge has representations

Figure 15. Statement 2: Knowledge representation can be explicit or implicit

Figure 16. Statement 3: KMS stores explicit representation of knowledge
The network (directory) model allows a user to locate a knowledge source by pointing to it. Galuppe (2002) mentions that KM dictionaries point at knowledge sources such as people, documents or databases. An example is organizational knowledge maps that enable locating knowledge or individuals possessing knowledge (Offsey, 1997). Based on above, we suggest that KMS points at representation of knowledge (Figure 17).

![Figure 17. Statement 4: KMS points at representation of knowledge](image)

Borrowing the storage metaphor from individual-level memory processes, Walsh and Ungson (1991) present the notion of retention facilities (containers) of organizational memory. We reflect this in Figure 18.

![Figure 18. Statement 5: Implicit representation of knowledge are contained in containers](image)

The notion of containers can be classified into two types: *individuals or agents* and *organizational transformers* (Walsh and Ungson, 1991). Individuals retain information based on their own experiences and store knowledge in their own capacity. Transformers are ways (and mechanisms) by which an input can be converted to an output and the logic underlying the transformers can be considered embedded knowledge. Davenport and Prusak (1998) mention that knowledge often becomes embedded not only in documents or repositories but also in organizational routines, processes, practices, and norms. Recall, our definition of knowledge is
based on actions that can be used to change states. This fits well with the ideas of transformers as knowledge enables transformation to occur. We describe the two types of containers in Figure 19.

Figure 19. Statement 6: Containers can be agent and organizational transformers

We summarize the above discussion in the propositions listed in Table 2.

| • Knowledge has representations       |
| • Representations of knowledge can be explicit or implicit |
| • KMS stores explicit representation of knowledge |
| • KMS points at representation of knowledge |
| • Implicit representations of knowledge are contained in containers |
| • Containers of implicit knowledge are of two types: agents and organizational transformers. |

Table 2: Statements underlying a conceptual model of KMS

We present these statements together in Figure 20 as a conceptual model of KMS.
As for the case of the previous models, we note that the novelty of this model is not in the individual statements but in combining them into one model.

### 2.4 Uses and Implications of the Models

In this section we discuss some possible uses of the models. We first demonstrate the application of the concepts with an example. We then suggest application to knowledge identification, tagging, and queries. Next, we use the models to differentiate information and knowledge and explain the differences and links between IS and KMS. Finally, we suggest some implications to the design of IS.
2.4.1 An Example of Using the Constructs

We demonstrate the constructs used in the models of knowledge via an example. Consider an applicant for admission to a university. An admission committee reviews the application and decides on accepting the applicant, rejecting the applicant, or withholding the application due to incomplete information. The possible agents in this example are: the applicant, committee members and the university administrators. However, if we deal with decisions within the university, the relevant agents for analyzing the required knowledge do not include the applicant. In this case, the application can be an entity whose state requires a change from “awaiting a decision regarding admission” to “provide with a decision regarding admission.” The possible actions to be taken by agents (admissions committee members) might include assembling information (such as quality of prior applicants), obtaining opinions, and deliberating the merits of an applicant. The outcome of executing these actions will be to change the state of the application (from “pending” to “decided”). In this case, knowledge is the ability of the admission committee to select actions in order to reach a decision regarding the application. In Table 3 we show how the elements of this case can be mapped into the concepts of the knowledge model.

<table>
<thead>
<tr>
<th>Knowledge related constructs</th>
<th>Example Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Committee members, Chair</td>
</tr>
<tr>
<td>Environment of the agent</td>
<td>Applicant, University administration</td>
</tr>
<tr>
<td>Resource (Non-Agent)</td>
<td>Admission application, Admissions manual</td>
</tr>
<tr>
<td>Entity (whose state needs to be changed)</td>
<td>Admission application</td>
</tr>
<tr>
<td>Current state of affairs</td>
<td>Application pending</td>
</tr>
<tr>
<td>Actions</td>
<td>Assemble information, obtain opinions, discuss application, rank-order applications, obtain more information</td>
</tr>
<tr>
<td>States</td>
<td>Pending admission application, Application under review, Decided application</td>
</tr>
<tr>
<td>Goal</td>
<td>Decided admission application</td>
</tr>
</tbody>
</table>

Table 3: Examples of the knowledge related constructs

2.4.2 Knowledge identification, tagging, and queries

Knowledge managed via a KMS needs to be identified so it can be searched and located. To do this, the knowledge has to be organized in clearly defined “chunks.” Zack (1999, p. 48) defines a knowledge element as “an atomic packet of knowledge content that can be labeled, indexed,
stored, retrieved and manipulated. Such packets may include processes, actions and events.” This
definition, however, does not indicate how such elements might be recognized, and “marked” or
“tagged” so that they can be organized and accessed. Moreover, in a given situation, several
elements might be needed to form the relevant knowledge “chunk”. We suggest that the concepts
we used to define knowledge, namely states, actions and goals, can be used to label and index
knowledge elements (instances). Furthermore, the view that knowledge has to do with choosing
actions in a context points at a way to partition knowledge into meaningful “chunks”.
Specifically, a chunk can involve the guidance for reaching from a given state to a goal state as
being a “chunk” or an “element”. This element can be tagged by describing the situation as well
as the desired outcome. We make two notes here. First, the content of the “chunk” depends on
the actions that might be potentially available to the knowledge seeker. Hence, chunks might
differ depending on the role of the involved agent. For example, a nurse dealing with a patient
emergency might have different choices of actions than a physician. Second, once knowledge is
partitioned into “chunks” which are identified and tagged, it can be queried. A user can form a
query by mentioning an initial state and a goal state of an entity to obtain a set of available
actions.

2.4.3 On the Distinction between Knowledge and Information
The KM literature presents several contrasting views of the relationship between knowledge and
information. We discuss this issue as the distinction might help to differentiate KMS from IS.
Some researchers ((Bhatt, 2001; Davenport, 1997; Hicks et al., 2002) consider the link between
information and knowledge to be “linear” in that knowledge is an “evolution” of information.
Others (such as Preiss, 1999; Tuomi, 2000) mention that this evolutionary view is problematic
because it makes it difficult to distinguish between information and knowledge. According to
another view, new knowledge is created from the acquisition and accumulation of information,
thus information can be viewed as providing necessary building blocks of new knowledge
((Malhotra et al., 2005; Nonaka, 1994). A contrasting view suggests that there is no definite
requirement for information to be available in order to create knowledge. Rather, new knowledge
is directly generated from existing knowledge (Li and Kettinger, 2006).
The above differing views point at a need to clarify the relationship between information and knowledge in the context of KMS. We suggest that our proposed conceptual models can be used for this purpose. We first observe that according to the models the link between information and knowledge is "non-linear" in that knowledge does not have to evolve from information. According to the models, knowledge is the ability to select actions to accomplish a goal, when the state of affairs is given. Thus, to know which knowledge is required, an agent needs to first identify the current state, goal state and available actions. We suggest that all these aspects of the situation can be considered information. In particular, the suggestion that the current states of affairs be considered information agrees with the view of an IS as a representation of a domain in the "real world" (Wand and Weber, 1993). Moreover, (Rolland et al., 1999) suggest that the context of knowledge access can be defined in terms of current and desired situations. In this respect, we claim that information helps agents recognize the context in which the knowledge is sought. The perceived state might affect the selection of actions. Thus, information is used to provide context for the needed knowledge and to help identify the required knowledge. This view is in agreement with Weinberger (1998) that information can help the agent identify actions, perceive alternative actions (such as by identifying an action that the agent was not aware of), create a selection mechanism for action (such as rule or preference), and bias the agent to select particular actions. However, these activities do not relate to the nature of the actions, but rather to the ability to identify actions. In this vein, Weinberger (1998) mentions that factual information is insufficient for selecting actions. Rather this information will help understand the situation (or context) based on which actions are to be taken. This information, in turn, can help an agent develop understanding of the outcomes of possible actions.

We note that some actions rather than changing the state of the environment might change the state of the agent. In particular, these actions might involve information processing activities necessary for the agent to better understand the current state and choose actions accordingly. For example, a manager rewarding salespeople will go from a state of "no decision" to "reward decision" based on sales information. This action will only involve internal processing of that information. The recognition that this action is needed to accomplish a goal is knowledge for that agent. Moreover, even the actions involved in processing the information
might be part of the knowledge. For example, when considering job applicants, the rules on how to rank the applications would specify actions needed to reach the ranking. In that respect, this is also knowledge. In this case, the knowledge indicates how the internal state of an agent can be changed.

2.4.4 Differentiating the roles of IS and KMS

The proposed distinction and relationship between information and knowledge in the context of KMS can help define the differences between IS and KMS. The role of an IS is to help an agent become aware of a situation. This situation involves the current state, desired state and the actions that are available to the agent. In contrast, the role of a KMS is to enable the agent to choose actions that can be taken in this situation.

More specifically, KMS are intended to be used by agents to: (a) help access descriptions of recommended sets of actions that are represented explicitly, or can be learned by accessing implicit representations of these actions; and (b) help the agent choose the actions in a particular situation. In comparison, an IS provides agents with the context in which knowledge is required, in terms of current and desired states, and available actions.

Another role of IS in the KMS context can be identified. Recall Weinberger’s (1998) suggestion that information can help agents select actions. To select actions, an agent will need to know their potential outcomes. In our models, the effect of an action is described in terms by the change of state it can bring about. Considering an IS as a state tracking mechanism (Wand and Weber, 1993), it follows that an IS can provide a record of past actions and their outcomes. If past states can be related to actions, the agent might be able to infer the possible outcome of actions in a given context. Such analysis is typically supported by Business Intelligence (BI) tools.

The above analysis leads to two conclusions. First, KMS and IS have complementary roles with respect to knowledge management context. Second, IS can provide this support via information about current context and about the past.

Another requirement for the IS in the KMS context can be identified. Hahn and Subramani (2000) suggest that a repository of information presented to an agent for selecting actions may not be helpful to the agent for two reasons. First, the agent may not able to identify...
the relevant (or useful) information from the repository. Second, if the relevant information is not presented in a form enabling the agent to relate it to the problem situation, the information cannot be used. Accordingly, another requirement is that an IS presents information in a form that can help agents select actions.

As mentioned in the introduction, several software products have been proposed as KMS. Maier (2002, p. 74) mentions, “many authors use the terms knowledge management tools or knowledge management systems to describe systems with quite similar intentions and functions. So far, there has been no clear distinction between these two terms.” He cites some examples of KM tools: Document and Content management, Intranet, Groupware, and Business Intelligence (Maier, 2002). We suggest that the conceptual models can help clarify the role of these KM tools. Using the distinction between IS and KMS discussed above, these KM tools can be differentiated according to the role they play in KMS. Whether a software tool will be classified as an IS or KMS will depend on what content it manages. Specifically, if a tool provides descriptions of the actions (such as organizational procedures and best-practices) that can be taken to accomplish certain goals, then it acts in a KMS role. It is possible that such tool will also provide explanations of the possible outcomes. It follows that tools such as Groupware, Document Management Systems and Intranet can be considered KMS if they are used to provide lists of possible actions or point out to resources where such lists can be found. On the other hand, a tool might just provide information that can be useful for the agents to become aware of the state of the environment. For such cases the tool will be used in the role of an IS. A particular type of tool – Business Intelligence might be considered as KMS if it is used to help users analyze past activities to predict the possible outcome of actions for a particular situation.

2.4.5 Guidelines for KMS Development

Frameworks based on philosophical theories/concepts have been used to suggest KMS design principles. Based on “hermeneutic circle of understanding” (Gadamer, 1975), Butler and Murphy (2007) propose a set of design principles for developing KMS. Their principles focus on assisting social actors to record their difficulties in understanding phenomena that they encounter in their work. In particular, they suggest that KMS should allow supporting actors in interpreting and understanding “whole-part relationships” of a domain. Richardson et al. (2006) suggest
KMS design principles based on Churchman’s inquiry theory (Churchman, 1971) where inquiry is defined as the process of creating knowledge, Habermas’ theory of communicative action and discourse ethics (Brocklesby and Cummings, 1996). The suggested principles focus on the need for accounting for ethical and moral principles in designing KMS. However, we note that these frameworks do not consider what knowledge is in the KMS context. Hence, the proposed guidelines developed from these frameworks do not attempt to identify or capture knowledge as managed by KMS.

Shanks et al. (2003) indicate that a conceptual model can be used to facilitate the design and implementation of an IS. We now suggest how our proposed conceptual models can be used to guide the development of KMS. In particular, this guidance relates to roles supported by a KMS, to knowledge identification, to knowledge tagging and to knowledge querying.

**Identifying role**

As mentioned above (section 2.3.1, p. 15), agents act in given roles. The role indicates the responsibilities of and actions available to the agent (consider the example of physician and nurse in section 2.4.2, p. 31). The responsibilities can be interpreted as the situations in which the agent needs to act. The actions will be chosen from those the agent in the given role can perform. Since the definition of knowledge is based on situations that call for actions and choice of actions, we contend that identifying roles would be the first step in analyzing the requirements for a KMS. Hence, we propose:

**Guideline 1:** Identify the roles for which the KMS is designed.

**Identifying knowledge**

A systems analyst defining the requirements for a KMS can ask users to identify contexts in which knowledge might be sought. Such contexts could be specified in terms of situations calling for actions, and lists of possible actions available to the users. Specifically, the situations can be described in terms of current and desired states. We therefore propose the following guideline for KMS design:

**Guideline 2:** For each role, identify the contexts in which knowledge might be sought

(1) List situations that users in a given role might need to act in terms of current and desired states
(2) List possible actions available to the users who act in this role

**Tagging knowledge**

To tag a knowledge element, situations can be described in terms of current and desired states. Such descriptions can be represented in informal language or by using a given set of concepts. Such sets of concepts and their relationships are termed ontologies, and when formalized they are called formal (or IS) ontologies (Gomez-Perez et al., 2004). The conceptual models even suggest that such ontology might be organized around the specific types of agents (i.e. roles), actions, and situations in the given domain.

**Guideline 3:** Consider ontologies as a way to describe situations

(1) A specific ontology can be developed for a given knowledge domain using the concepts of roles, states, and actions.

(2) The ontology that describes a specific situation can be created by instantiating the knowledge related concepts.

An example of such a sample ontology is shown graphically in Figure 21. It was developed using concepts from the Web Ontology Language – OWL (McGuinness et al., 2004). The diagram depicts the classes of the ontology, where each class represents a concept of the ontology. The rectangles represent the classes (i.e. knowledge related concepts) and the arrows indicate the relationships between the concepts (solid white for class subclass hierarchy and solid black for classes related to other classes). The properties of these classes are indicated inside the rectangles.
Querying the KMS

The notion of knowledge as a guide to choosing actions in a given context, also points out at the way queries might be posed to the KMS. A query may be phrased using terms similar to the ones used for tagging knowledge instances. The results of a knowledge query (how many responses are obtained) will depend on how specific a query is framed, i.e. how much detail it provides in describing the situation, goals, and the available actions.

To provide an example on how the guidelines can be implemented we use the admission case described above (Section 2.4.1). In this scenario a situation could be: current situation is "application needs to be processed" and the goal would be: "reach a decision about the application." The possible actions might be determined based on the role of the expected user. For example, an admission clerk might have different actions available than to the admission committee chair. An ontology for this situation has been created by instantiating the knowledge related constructs (Figure 21). This ontology is shown in Figure 22. For simplicity the diagram does not show the effect of actions on states.
Using this ontology, a user can form a knowledge query by indicating the current state and the desired state. The actions specified in the query can be limited to a given set. If the actions are not specified, all possible ways to accomplish a goal given the situation might be provided by the KMS.

2.5 Comparing the Models to Related Work

In this section we relate the proposed models to the KM literature.

2.5.1 Comparing to the Learning-Oriented KMS Model

With one exception, we could not identify conceptual models of KMS in the literature. The exception is the Learning-Oriented KMS (LOKMS) model ((Hall and Paradice, 2004; Hall et al., 2003)). The model combines principles from inquiring systems (Churchman, 1971) with Simon's intelligence-design-choice decision making model (Simon, 1955). The purpose of the model is to describe systems to enhance organizational decision-making, knowledge creation, and
knowledge management. The model is based on three sequential phases: information acquisition and hypothesis generation (intelligence phase), knowledge creation (design phase), and decision support (choice phase). The intelligence phase comprises the activities: identify actions necessary to update the existing knowledge base, detect opportunities or needs to solve problems, develop hypotheses regarding relationships of newly discovered information, and define a desired state that may be a goal. The knowledge creation phase involves analysis of the desired states and the hypotheses. The desired states and the hypotheses are then added to the knowledge store as new knowledge. In the decision phase a decision maker uses the results of the knowledge creation phase combined with his/her tacit and experiential knowledge to choose actions that will change the current state to the desired state.

We note that the authors of LOKMS, similar to our approach, consider that actions can change a current to a desired state. However, we note two differences between LOKMS and our proposed models. First, LOKMS provides a process view on how organizations acquire and use knowledge. However, they do not define knowledge. In contrast, our proposed models focus on defining knowledge in the context of KMS. Second, LOKMS considers both the knowledge and information as stored in a common base. This model does not provide a clear distinction between knowledge and information. Thus, the role of a KMS is not distinguished from that of an IS. In contrast, the proposed conceptual models differentiate knowledge from information. There is one more distinct difference in the approaches. The LOKMS authors claim: "..."choice" [of action] ultimately resides with the user. Consequently, the third phase of the system, Choice, does not have a technological component" (Hall and Paradice, 2004, p.5). In this respect, their proposal is akin to a decision support system. In contrast, our models define knowledge as the ability to make the choice, and view a KMS as a way to provide the user with this knowledge (or point at this knowledge).

2.5.2 Analyzing other Definitions of Knowledge

In the absence of any "real" competing framework that addresses the issue of defining knowledge in the context of KMS, we analyzed a set of 26 definitions of knowledge in the KM literature (Table 17, Appendix A). Each knowledge definition was coded for its use of the concepts: agent, action, state and goal (0 for absence and 1 for presence of a keyword or
synonym implying the concept). We observed that the word “information” appeared frequently in the definitions and therefore we treated information as a concept for this analysis. We counted the frequency of the concepts (or their combinations) in the definitions. The notions of agent and action appeared in almost half of the definitions (35% and 50% respectively). However, together these concepts appeared in only four definitions (15%). Surprisingly, the notion of current state of affairs appeared only once. It is interesting also to note that about 60% of the definitions viewed knowledge as information. Finally, not a single reference combined the concepts of agents, actions, states, goals and knowledge in one definition. The details of the combination of the knowledge related concepts are mentioned in Table 18 (Appendix A).

In addition to information, we identified three common concepts used in the analyzed definitions that we have not included in our models. These concepts are “context” (7 of 26), “belief” (4 of 26) and “experience” (7 of 26). We briefly discuss how these concepts can be related to the proposed models. The frequency of other concepts (not considered in our models) is provided in Table 19 (Appendix A).

The term “context” has been defined in the literature as: “a new view of the world” (Nonaka and Takeuchi, 1995, p. 42) and “systemic relations between individuals and the environment” (Jensen, 2005, p. 54). In our model, the view of the world is provided via the notion of states of relevant entities in the environment (Figure 10). The term belief is defined by Bratman (1987, p. 15) as “some aspect of the agent’s knowledge or information about the environment, self or other agents.” As indicated above (section 2.3.1, p. 22) in our models beliefs reflect what the agent thinks the outcomes of actions will be. Thus, our notion of belief is more specific than that of Bratman. Finally, we did not find a definition for the concept “experience” in the context of knowledge, so we could not compare it to the concepts in the proposed models.

2.5.3 Comparing to Knowledge views

Several views of knowledge can be found in the KM literature. Alavi and Leidner (2001) summarize these views into five types (Table 4).
Knowledge view | Definitions
---|---
State of mind | “Knowledge has been described as “a state or fact of knowing” with knowing being a condition of understanding gained through experience or study. The perspective on knowledge as a state of mind focuses on enabling individuals to expand their personal knowledge and apply it to the organization’s needs.”
Capability | “[K]nowledge can be viewed as a capability with the potential for influencing future action. ...knowledge is not so much a capability for specific action, but the capacity to use information; learning and experience result in an ability to interpret information and to ascertain what information is necessary in decision making.”
Process | “[K]nowledge can be viewed as a process of simultaneously knowing and acting. The process perspective focuses on the applying of expertise.”
Object | “[K]nowledge can be viewed as a thing to be stored and manipulated (i.e. an object).”
Access to information | “[O]rganizational knowledge must be organized to facilitate access to and retrieval of content. This view may be thought of as an extension of the view of knowledge as an object, with a special emphasis on the accessibility of the knowledge objects.”

Table 4: Knowledge views (Alavi and Leidner, 2001, p. 110)

We first note that these views can be classified into two groups: knowledge definition (first three) and knowledge representation (last two). We find that our models can be compared to only the second, fourth, and fifth views. The view of knowledge as a capability is closely related to the definition of knowledge from an agent’s perspective (Figure 12) as the ability to choose actions in order to change the state of affairs. The main difference is that the definition of Alavi and Leidner refers to information selection and processing actions. In contrast, in the proposed models actions could lead to any change in the state of the agent or its environment and are not limited to only to the agent’s states and only to information processing.

The view of knowledge as an object to be stored and manipulated corresponds to the conceptual model of knowledge as an object of use (Figure 13). In this model, knowledge is treated as an entity that can be stored and manipulated.
2.6 Conclusion and Future Work

This paper presented a framework for knowledge in the context of KMS. The framework is based on two foundations. First, that knowledge is related to action. Second, that knowledge can be formalized using the concepts of agents, actions, states and goals. The relationships among the concepts were identified by analyzing the relevant literature and served as the basis for developing conceptual models for KMS. Three models were formulated: first, knowledge as possessed by an individual agent; second, knowledge as used by an agent (not necessarily the one who originally created it); and third, knowledge as managed in a KMS.

The possible value of the models is demonstrated by discussing some implications and possible uses. In particular, the models serve to show the differences between information and knowledge and IS and KMS. As well, the models serve to show the link between IS and KMS.

The models are also used to suggest guidelines for the design of KMS. These guidelines address, in particular, for the identification of user roles and knowledge needs, for tagging knowledge instances, for querying a KMS, and for the application of formal ontologies in KMS.

A clear next step is to test the feasibility of the development of the models. The implications described above hinge on the description of knowledge in terms of a quadruple: agent, current state, goal state and possible actions. For a given or unspecified agent, the knowledge can be viewed in terms of the last three. This view can be considered a proposition that can be tested. More specifically, the description and tagging of knowledge in terms of current state, goal state and possible actions can be tested for usability and usefulness. In the following we briefly suggest some possible tests.

1. Experts in a given domain can be asked to describe in free form the way they suggest to solve a certain problem in the domain. Several coders can then be asked to describe the generated statements in terms of current states, goal states and actions proposed.

2. Experts can be asked about the solution to a given problem in interviews guided by questions about current states, goals, and possible actions. The descriptions obtained this way can be compared to descriptions obtained without guiding the experts.
(3) Documents describing organizational procedures can be analyzed to see if the four (or three-excluding agents) elements of knowledge can be identified by trained coders. The results provided by different coders can be compared for consistency.

(4) Subjects searching for knowledge (e.g. in a collection of documents or on the Internet) will be asked to form knowledge queries about given issues. Some subjects will be trained to frame the issues in terms of a given state, a desired state and possible actions. The quality of the queries (and outcomes) for these subjects will be compared to those generated by untrained subjects.

Another direction of research is to use the models in a wider context. The KM literature distinguishes knowledge possessed and used by an individual from knowledge acquired and shared by a group (Alavi and Leidner, 2001). To understand group knowledge processes, we need to develop a better understanding of the individual view of knowledge. This was the purpose of the present work. A next logical step would be to incorporate the proposed knowledge view in the analysis of knowledge processes.
References


Chapter 3
Using OWL as a Conceptual Modeling Language

3.1 Introduction

An Information Systems (IS) ontology can be defined as a formal, explicit specification of a shared conceptualization, which can be considered a simplified view of the world (Gruber, 1993). Formalized ontologies have been introduced in order to improve communication, share information and support reuse of IS components (Uschold and Gruninger, 1996). The emergence of the semantic web has created a special interest in formal ontologies as a way to represent semantics of web sources. In this context, Manola and Miller (2004) proposed the Resource Description Framework (RDF) - an infrastructure that enables the encoding, exchange, and reuse of structured metadata on the Web. Using RDF as a framework, ontology languages such as OWL (Web ontology language) have been proposed for creating formal descriptions of terminologies used in web documents (McGuinness et al., 2004).

Since formal ontologies comprise definitions of concepts and their relationships, they can be used to represent information about domains of interest (sometimes called “real world” domains, even when referring to non-existent situations). However, the focus of formalized ontologies has been on their use for computational purposes, not on their representation abilities. The purpose of this paper is to show how a formal ontology language can be used to create conceptual representations of domains as perceived by individuals or groups. Such representations are often termed Conceptual Models (Mylopoulos, 1992; Wand and Weber, 2002). We suggest a set of rules for using the OWL (Web ontology language) to construct conceptual domain models. We motivate the need for such rules by discussing some difficulties that may arise when representing domain concepts using OWL. We base our modeling rules on a philosophical ontology, i.e. a description of reality (Smith, 2001). Smith describes the difference between philosophical and IS ontologies. While the goal of the former is to establish truth about

---

A version of this chapter is submitted to IEEE Transactions on Knowledge and Data Engineering for publication.
reality, the latter are software (or formal language) artifacts designed with a specific set of uses and computational environments in mind (Smith, 2001).

The rest of the paper is organized as follows. Section 3.2 provides the background of this research. In this section, OWL is introduced and some difficulties in modeling general domain concepts in OWL are discussed. Section 3.3 discusses the possible benefits of using philosophical ontologies to derive modeling guidelines. This section introduces Bunge's ontology and mentions some alternative ontological approaches. Section 3.4 provides specific suggestions on using OWL concepts to represent domains, followed by examples to illustrate the suggestions. Section 3.5 suggests how to model interactions in OWL. Examples and a case are used for demonstration. Section 3.6 concludes with a summary and a discussion of further possible research.

3.2 Background

Conceptual modeling is the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication (Mylopoulos, 1992). Among several uses of conceptual modeling, the most common are: (1) facilitate communications between users and modelers, (2) increase analysts understanding of the domain and (3) serve as the basis for design and implementation (Wand and Weber, 2002). Conceptual models are created using modeling languages that provide constructs for representing real-world phenomena and rules for combining these constructs (Shanks et al., 2003). Since theories of ontology can be used to represent phenomena of a given domain (Shanks et al., 2003; Wand and Weber, 2002), we suggest that such theories can guide the use of ontology languages for conceptual modeling. Prior to discussing how this can be done, we first discuss some difficulties of modeling domain concepts in OWL.

3.2.1 OWL

OWL (McGuinness et al., 2004) has been created by the W3C ontology working group to enable publishing and sharing IS ontologies on the web. OWL is currently considered one of the key Semantic Web technologies that provide a framework for data sharing and reuse on the Web (Gomez-Perez et al., 2004). OWL provides for defining classes, individuals, properties of classes
and individuals and assertions about these properties. Further, OWL allows reasoning about classes and individuals (based on its formal semantics). OWL is divided into three layers of increasing level of expressiveness: OWL Lite, OWL Description Logic (DL) and OWL Full (McGuinness et al., 2004). In this paper we refer to OWL DL unless specified otherwise. The key OWL concepts are summarized in Table 5.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>A class is defined by a name and a collection of properties that describes a set of individuals. Individuals belong to a class and inherit properties of the class. All classes are subclasses of the class OWL:Thing.</td>
</tr>
<tr>
<td>Individuals</td>
<td>An individual exists in the “universe of things” and is “minimally introduced” by declaring it to be a member of the class OWL:Thing.</td>
</tr>
<tr>
<td>Properties</td>
<td>Properties assert general facts about members of classes and specific facts about individuals. Several mechanisms are used to specify different types of properties such as transitive, symmetrical, inverseof and functional.</td>
</tr>
</tbody>
</table>

Table 5: Concepts of OWL (adapted from (McGuinness et al., 2004))

3.2.2 Difficulties in representing real-world concepts in OWL

OWL is intended to provide representation of terminologies using, classes, instances and properties. No clear guidelines exist for using OWL constructs to properly represent the semantics of a modeled domain. Several technical papers (mostly online) discuss guidance on developing ontologies in OWL. For example, Horridge et al. (2004) and McGuinness et al. (2004) explain and suggest use of different OWL constructs to develop ontologies. However, these papers mostly provide suggestion on how to construct terminologies in OWL, not on how to use its constructs to convey domain semantics. To demonstrate this point, consider OWL classes. Mcguinness et al. (2004) propose that classes correspond to “naturally occurring” sets of things in a domain of discourse, and individuals correspond to actual entities that can be grouped into these classes. However, there is no requirement about the nature of the individuals or their existence. The phrase “naturally occurring” does not reflect things as they exist in real world. Any concept from a domain of discourse can be considered an individual or a class in OWL.

---

5 We consider OWL-Full, while providing examples in this section. In OWL-DL the same name cannot be used for a class, a property, and an individual.
It follows, that users of OWL are allowed substantial freedom and flexibility (within the OWL syntax rules) in how they can apply OWL basic constructs to represent phenomena in the domain of interest. This is advantageous as the language is flexible, universal and can be applied to multiple domains. However, we claim that such freedom, together with the lack of modeling guidelines and constraints, may become a drawback if the purpose is to use OWL for conceptual modeling. First, it might lead to inconsistent domain representations, as the same phenomenon might be represented in different ways. This, in turn can cause ontologies to become “unstable” in that they might require significant changes when more domain knowledge is acquired. The problem might affect applications depending on the information contained in the ontology. Second, as no clear rules exist to map from domain information to OWL constructs, it might be difficult to infer the original information from a given ontology. Third, some general phenomena about the real world domain may be difficult to represent in ontologies created by OWL, leading to incomplete representations. We discuss the two difficulties in the following subsections and address the third in section 3.5.

1. Inconsistent representations

As OWL does not provide guidelines on how to map domain phenomena into ontological constructs, it might not be clear whether a certain domain fact should be modeled as an individual, a class, or a property. Hence, different modelers might represent the same domain phenomena differently. We term the possibility of using different OWL concepts to represent the same fact ontology inconsistency. The example below (adapted from (McGuinness et al., 2004)) shows how an individual could be represented also as a class.

```xml
<owl:Class rdf:ID="WineGrape"/>
<WineGrape rdf:ID="CabernetSauvignonGrape"/>
```

Here CabernetSauvignonGrape is an individual of the class WineGrape denoting the actual grape variety called Cabernet Sauvignon. However, as shown below, CabernetSauvignonGrape can also be considered a class - the set of all Cabernet Sauvignon wine grapes produced by a specific winery in a particular year (indicated by instances such as CabernetSauvignonGrape1).
An OWL property can also be represented as an individual or a class. Consider an object property madeFromGrape which relates the Wine class to the WineGrape class. This is shown in (McGuinness et al., 2004) as:

```xml
<owl:ObjectProperty rdf:ID="madeFromGrape">
    <rdfs:domain rdf:resource="#Wine"/>
    <rdfs:range rdf:resource="#WineGrape"/>
</owl:ObjectProperty>
```

The property madeFromGrape can also be shown as an individual or a class as below:

```xml
<owl:Class rdf:ID="madeFromLiquid"/>
<owl:Class rdf:ID="madeFromGrape"/>
<owl:Class rdf:ID="madeFromGrape2"/>
<owl:Class rdf:ID="madeFromPinotNoirGrape"/>
<owl:Class rdf:ID="madeFromCabernetFrancGrape"/>
```

The above example of representing property as a class or an individual may look unnatural to OWL users but the intention here is to demonstrate that a given concept can be represented in OWL as an instance, a class or a property.

One explanation for the multiple ways in which a fact can be represented in OWL is that the notions of individual and class are borrowed from database design where an individual record in the database is considered an instance and collection of such instances is considered a class. It is not necessary that an individual record of a database represents a real world instance.

Due to the problem of inconsistent representation, the way domain phenomena are represented might change over time, as more domain knowledge is acquired. For example, in the Wine ontology, if CabernetSauvignonGrape is first modeled as an individual in the class GrapeWine, it will be difficult to model CabernetSauvignonGrapes produced in a specific year by a specific winery at a later stage. Since there are several instances of CabernetSauvignonGrape that can be classified by the year of production, the concept CabernetSauvignonGrape would no longer be an individual but a class. This example indicates how the instantiation of the ontology concepts may have to be changed as more is learned about the domain.
Noy and McGuinness (2001) suggest that the best way to create a domain ontology almost always depends on the expected application and anticipated extensions. However, it is often not easy to anticipate how the ontology will be used or extended in the future. While it might be relatively easy to introduce a new concept into an existing ontology, it likely will be much more difficult to change the representation of domain concepts that have already been instantiated (e.g. changing an individual to a class). The problem might occur because OWL concepts such as class and individuals have not been assigned a well-defined ontological semantics.

2. Difficulty in interpretation
Because no clear mapping guidelines exist from domain phenomena to OWL constructs, the specific constructs used in a given ontology might not provide sufficient cues to determine the original intention. Thus, interpretation of domain facts represented in OWL may be difficult at times. For example, when we come across a class Wine in an ontology described in OWL, it is not immediately clear whether the ontology refers to the general concept of wine or to a specific bottle of wine. To understand the semantics carried by the Wine class, one has to carefully look at the individuals and properties of the Wine class and apply additional domain knowledge. For example, a property of Wine class such as Barcode might indicate that the OWL concept Wine refers to a specific bottle of wine and not the general concept of wine.

We suggest that above difficulties would be alleviated if clear guidelines existed for mapping domain phenomena into OWL constructs and back. The creation of such guidelines requires that a set of possible domain phenomena be defined. We suggest this set to be determined based on a philosophical ontology. In the next section, we describe the use of such an ontology, which takes a realist position.

3.3 Using philosophical ontologies
In recent years IS researchers have realized that they are facing a problem of establishing a relationship between representation of reality and reality itself - the problem that philosophers dealt with over centuries (Smith, 2001). This has led to the recognition that the theory and practice of IS can benefit from ontological theories (Milton, 2000; Weber, 2003). In philosophy,
Ontology is the branch of philosophy that deals with the order and structure of reality in the broadest sense possible. Some examples of philosophical ontologies applied in the context of IS are Bunge's ontology (Bunge, 1977), Sowa's ontology (Sowa, 1984), and Chisholm's ontology (Chisholm, 1996).

Several efforts have been conducted on inculcating IS theories with philosophical ontologies. For example, Milton proposes a unifying framework for data modelling using Chisholm's ontology (Milton, 2000) and Wand and Weber (1990b) propose the use of Bunge's ontological theory to provide foundations to IS analysis and design. Specifically, Bunge's ontology has been used to analyze IS conceptual modeling methods and to generate predictions about their efficacy (Green and Rosemann, 2000; Wand and Weber, 1993). More recently, Guarino and Welty (2002) have used philosophical notions such as identity and part-whole relations to identify inconsistencies in formal ontologies.

We use concepts from Bunge's ontological work (Bunge, 1977) to prescribe guidelines for representing domain semantics in OWL. There are several reasons for selecting Bunge's ontology. First, Bunge's ontology has been used to distinguish instance, class, and properties (Parsons and Wand, 1997). Such distinction is crucial for prescribing guidelines to model real-world concepts. Second, Bunge's ontology (which takes a realist position) deals with real-world concepts and thus fits the objective of the paper of providing real-world semantics to an ontology description language. Third, Bunge's ontology is well formalized, comprehensive and has been adapted and extended for information systems (Wand et al., 1995; Wand and Weber, 1990a, 1993; Weber, 1997). In particular it has been used to generate theoretical predictions about conceptual modeling techniques and languages (such as ER (Wand et al., 1999), NIAM (Weber and Zhang, 1996) and UML (Evermann and Wand, 2005)). Moreover, several of these predictions have been empirically examined and corroborated (such as (Burton-Jones and Meso, 2006) and (Gemino and Wand, 2005)). In the next section, we briefly present the main concepts and premises of Bunge's ontological view.

Prior to describing the main concepts of Bunge's ontology we note that although these concepts seem to reflect a realist positions (i.e. that an "objective reality" exists) we use them as guidance for attaching semantics to OWL concepts. There is nothing to proscribe the use of
these concepts to model imagined (or constructed) reality. Furthermore, it can be justified on the grounds that evidence exists that the way humans store concepts is affected by their interactions with their environment (Martin, 2007).

3.3.1 Bunge's Ontology

Bunge's ontology (1977) includes a set of high-level constructs and specifies certain relationships of things, properties and classes. Bunge defines the world as the aggregation of its constituents, which are entities or substantial things and states that:

- **All things have properties** (which exist whether humans are aware of them or not)
- **Properties are always attached to things** (“every property is possessed by some individual or other; there are no properties that fail to be paired to any individuals” (Bunge, 1977, p.62).
- **Classes and kinds are secondary to things and properties – they are defined in terms of properties.** A class in Bunge's ontology is defined by a set of things possessing a common property and a kind is defined by a set of common properties.

Bunge distinguishes substantial things (which exist spatio-temporally) from conceptual things which include “Whatever is general is either a property (e.g. a law) or an attribute (in which case it may be called a universal) or a proposition or a set of propositions (e.g. a theory)” (Bunge, 1977 p., 157). Properties can be intrinsic or mutual – namely belonging to several things. Properties of conceptual entities are called attributes. In particular, a thing can be modeled in terms of a set of attributes, which are characteristics assigned to things by humans. The main concepts of Bunge's ontology are described in Table 6.

---

6 Bunge uses his concepts also to represent social and cultural systems. He comments that his view should not be "construed in the narrow physical way." (Bunge, 1979).
<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thing</td>
<td>A thing is the elementary unit in the ontological model. The world is made of things. A distinction is made between concrete things (e.g. a book) and conceptual things (e.g. a mathematical set). It is assumed that any domain can be described by concrete things and the linkages between them.</td>
</tr>
<tr>
<td>Properties</td>
<td>Things possess properties. A property that is inherently a property of an individual thing is called an intrinsic property. A property that is meaningful only in the context of two or more things is called a mutual property. For example, height is an intrinsic property of a person and salary is a mutual property between a person and a company.</td>
</tr>
<tr>
<td>Composition</td>
<td>A composite is a thing that is made up of other things. Composites possess emergent properties — properties not inherited from their components. For example, a computer has a property “processing power” not possessed by any of its components.</td>
</tr>
<tr>
<td>Attribute</td>
<td>A property is modeled via an attribute function that maps a set of things into a set of values at a given time.</td>
</tr>
<tr>
<td>State</td>
<td>The state of a thing is the vector of values for all attribute functions in a schema of a thing at a given time.</td>
</tr>
<tr>
<td>Event</td>
<td>An event is a change of state of a thing. It is affected via a transformation (see below).</td>
</tr>
<tr>
<td>Transformation</td>
<td>A transformation is a mapping of a set of states into itself.</td>
</tr>
<tr>
<td>History</td>
<td>The history of a thing is the chronologically ordered states that a thing traverses. Example: history of positions of an employee over a period of time.</td>
</tr>
<tr>
<td>Interaction</td>
<td>Interaction is the ability of a thing to change the “history” (states traversed) by another thing. The two things are said to interact or to be coupled.</td>
</tr>
<tr>
<td>System</td>
<td>A set of interacting things such that for every bi-partitioning of the set, coupling exists among things in the two sub-sets.</td>
</tr>
<tr>
<td>Composition</td>
<td>The set of things in a system.</td>
</tr>
<tr>
<td>Structure</td>
<td>The set of interactions in a system.</td>
</tr>
<tr>
<td>Subsystem</td>
<td>A subsystem is a system whose composition and structure are subsets of the composition and structure of another system. Example: a procurement system is a subsystem of a system of inventory management.</td>
</tr>
<tr>
<td>Law</td>
<td>A (state) law is a restriction on the possible values of the components of a functional schema of thing or their combinations.</td>
</tr>
<tr>
<td>Class and kind</td>
<td>A class is a set of things that possess a common property. A kind is a set of things possessing more than one common property.</td>
</tr>
</tbody>
</table>

Table 6: Concepts of Bunge’s ontology (Wand and Weber, 1993)

3.3.2 Other ontological approaches

We now briefly compare Bunge’s main concepts to those of some other ontological approaches applied in the context of IS. First, we note what is termed the fact-based school by Hirschheim et al. (1995) in their analysis of data modeling approaches. They present a universe of discourse (UOD) made of entities that exist independently of others entities and can be concrete or abstract. Concrete entities are physical phenomena such as “a car” and abstract entities are entities without a visible physical appearance. Abstract entities are results of some functions or are formed by a relationship, such as car-ownership (a relationship instance between two concrete entities - a person and a car). As opposed to Bunge’s ontology, the fact-based approach enables entities to exist independent of others. However, we note that Hirschheim et al. refer to
database design, where domain dynamics and hence interactions of things are not considered. The important point of agreement is the view of the world (UOD) as made up of things, substantial or conceptual.

Chisholm (1996) posits that ontology is a categorization of what exists. The fundamental assumption made in his ontology is that there are attributes, some of these attributes are exemplified by entities (e.g. being a dog), some of them are unexemplified (e.g. being a unicorn) and some of them cannot be exemplified (e.g. being a round square). Sowa (2000) presents a top-level ontology to describe the world. His ontology is based on characterization of entities into several types (called primitives such as concrete and abstract). An entity is either a primitive or a combination of the primitives. The important issue is that all three ontologies recognize things or entities, their properties or attributes, and classes of things or entities.

Table 7 compares the ontologies on their views on entities, classes, and properties.

<table>
<thead>
<tr>
<th>Approaches</th>
<th>View on entities</th>
<th>View on classes</th>
<th>View on properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burge</td>
<td>The world is made of things.</td>
<td>Classes (or kinds) are defined in terms of properties of things.</td>
<td>Things possess properties. Human think about things in terms of attributes.</td>
</tr>
<tr>
<td>Chisholm</td>
<td>An individual is a discernible and a transient object. An individual does not have to be material or physical in nature.</td>
<td>Classes include individuals having certain attributes. Classes are developed from individuals through the exemplification of an individual's attributes.</td>
<td>Individuals are identified using attributes that only they exemplify. Attributes are used to restrict membership of classes.</td>
</tr>
<tr>
<td>Sowa</td>
<td>Entities are concrete or abstract. A concrete entity has a location in space-time. Not so for an abstract entity. Entities are categorized into three primitive types: independent, relative and mediating. An independent entity needs not have relationships to other entities. A relative entity must have some relationship to others; a mediating entity creates a relationship between two entities.</td>
<td>Categories are defined by conjunctions of primitives. There are constraints on how instances of categories are related to instances of other categories.</td>
<td>An entity is characterized by some inherent Firstness independent of any relationships it may have to other entities.</td>
</tr>
</tbody>
</table>

Table 7: Ontological approaches on Entity, Class, and Property
Guidelines for modeling instances, classes, and properties in OWL

In this section we suggest how to use ontological guidance to convey domain semantics in OWL, so that OWL constructs can be used for conceptual modeling. Specifically, we suggest two types of guidelines. First, we propose mapping rules between ontological concepts and OWL constructs. In other words, we attach specific meaning to these constructs. This might appear quite natural when dealing with concepts such as things, properties and classes. However, not all ontological concepts can be directly mapped into OWL constructs. We claim such concepts can still be modeled by following some specific guidelines on how to employ OWL. In the next section we demonstrate how one important aspect of application domains – interaction – can be represented using OWL constructs while still following the mapping rules assigning ontological meaning to the constructs. To develop the guidelines and rules we will employ the following concepts from Bunge’s ontology: things (substantial individuals), properties, classes, attributes, states, and interaction. Since we model domains as perceived by humans, we will not make a distinction between properties and their representations as attributes.

In our ensuing discussion, we distinguish between guidelines and rules. Guidelines are informal statements on how to use OWL constructs in principle. Rules provide specific details about the implementation of the guidelines.

3.4.1 Mapping guidelines and rules

The most fundamental concept in Bunge’s ontology is that of a thing. Bunge distinguishes between substantial things – things that are thought to exist in time and space (also called entities) and conceptual things (such as properties, attributes and propositions). In the following, we will use the word “substantial thing” to refer to any entity that might exist. All other concepts, unless otherwise stated, will be considered conceptual things and termed “non substantial.” We note, however, that a thing does not have to actually exist to be perceived as substantial. In other words, a thing can be imaginary, or agreed-upon by a group of individuals. What matter is that it is perceived as something that might physically exist.

Guideline 1: Substantial things should be modeled as OWL individuals.
OWL does not place any restrictions on what can be modeled as individuals. Thus, OWL individuals are often used to represent concepts other than substantial things, such as properties and property values. This situation may lead to the problem of construct overload - where an OWL instance could represent anything. Such overload can undermine ontological clarity of the resulting IS ontologies (Wand and Weber, 1993). While it is unrealistic in current OWL syntax to restrict the use of individuals only for modeling substantial things, we propose to alleviate this problem by using the following guideline for modeling instances in OWL.

**Guideline 2:** When using OWL constructs in conceptual modeling, it is necessary to distinguish OWL instances that represent substantial things from non-substantial things.

To implement this guideline in OWL we suggest declaring two upper-level classes (subclasses of the owl:Thing class) – one for representing all substantial things and the other for representing all other instances which do not represent substantial things. Since no thing can be a substantial thing and non-substantial thing simultaneously, these two upper-level classes should be declared as disjoint.

The following modeling rule implements the first two guidelines.  

**Modeling Rule 1:** An OWL-based conceptual model should include two disjoint upper-level classes:

1. **Substantial_Thing** class - the extension of this class will consist of all OWL individuals that represent substantial things.

2. **Non_Substantial_Thing** class - the extension of this class will consist of all OWL individuals that are used to represent non substantial things.

From the above rule it follows that:

**Corollary 1.** Substantial things should be modeled as OWL individuals that are instances of the class Substantial_Thing or its subclasses. OWL individuals used for other purposes should be made instances of the Non_Substantial_Thing class or its subclasses.

---

8 In practice, ontology development environments (such as Protégé OWL [(Protege, 2003)] could be set to modellers to define classes that are subclasses of both these upper-level classes (such classes would be inconsistent since they would not have any instances). Reasoning tools will be able to detect such inconsistencies.
Corollary 2. Any OWL class, all instances of which are intended to represent substantial things, should be made a subclass of the Substantial_Thing class.

Corollary 3. No OWL individual can represent both a substantial thing and non-substantial thing at the same time.

Corollary 4. Other OWL constructs (such as OWL properties) should not be used to represent substantial things.

If the above guidelines and modeling rule are followed, then ontological substantial things can be mapped to OWL individuals in the class extension of the upper-level class Substantial_Thing and, every OWL individual that is an instance of the class Substantial_Thing will correspond to an ontological substantial thing. Thus, OWL individuals in the class Substantial_Thing are assigned ontological semantics.

At this point, it might be useful to provide some guidance on how to distinguish substantial from non-substantial individuals. We first note that the guidelines and rules assume the distinction can be done. How this is done, is not a question of how to construct models, but rather of how to understand a domain prior to model development. However, the question is still of importance in the context of modeling. Hence, we discuss it briefly. First, we note that what is important is not the actual existence of things, but rather their perceived existence. Such a perception can exist in one’s mind, or shared by a community of individuals (Smith, 2001). Second, assuming a phenomenon is a “candidate” for being a substantial individual, one can apply the following test: ask whether the thing can be conceived as existing in time and space (Bunge, 1977 p., 157). When considering classes, the question would be: can one conceive of instances that (might) exist in time and space. Thus, both dogs and unicorn would be considered substantial things. However, consider a bank account. While existing in time, it does not exist in space. All that can exist in space is the record of the state (or changes) of an account on a piece of paper. This record, however, should not be confused with the account itself. The question might arise then – what is an account for the purpose of modeling. We address this issue later (when discussing interaction in section 3.5.2).

The ability to identify substantial individuals also points out how to identify some important types of non-substantial things. In particular, any statement about the properties of substantial
individuals or their changes will be a non-substantial thing. Thus, properties and attributes of individuals, changes of properties of individuals, and laws governing possible properties and attributes or their changes, will be considered non-substantial things. It follows that types of attributes (e.g. "color") will be represented by classes of non-substantial things.

We illustrate the implementation of the above guidelines and rule in OWL next.

```xml
<owl:Class rdf:about="#Substantial_Thing">
  <owl:disjointWith>
    <owl:Class rdf:ID="Non_Substantial_Thing"/>
  </owl:disjointWith>
</owl:Class>

<owl:Class rdf:about="#Non_Substantial_Thing">
  <owl:disjointWith rdf:resource="#Substantial_Thing"/>
</owl:Class>
```

The minimal representation of a substantial thing, for example, a specific person, John Smith, will require that it is represented as an OWL individual and an instance of the class Substantial_Thing: `<owl:Substantial_Thing rdf:ID="John_Smith">`.

Once we have outlined the guidelines for modeling things in general, we now propose general guidelines related to properties. According to Bunge "every property is possessed by some individual or other; there are no properties that fail to be paired to any individuals" (Bunge, 1977 p. 62). Classes and kinds in Bunge's ontology are defined in terms of properties. A class in Bunge's ontology is a set of things possessing a common property and a kind is a set of things possessing a set of properties (not contradictory to other ontological approaches – see Table 7). In OWL no syntactic restrictions are placed regarding the relationships between individuals, classes, and properties. Hence, no direct support is available in OWL to support the above ontological assumptions. Specifically:

- **An OWL individual does not have to possess at least one property**
  - In OWL, an individual can simply be introduced as an instance of some class or as a minimum, as an instance of the top class owl:Thing. The instance may not possess any properties associated with it (either directly or by virtue of class membership).

---

9 Since classes of things are not things, they will be considered conceptual things. The aggregate of instances of a class of substantial things is a substantial thing, but this is not the class.
• An **OWL class does not have to be associated with at least one property**
  - In OWL, properties can be associated with classes by using property restrictions, such as value or cardinality restrictions. However, OWL also allows defining named classes without using property restrictions.

• **OWL does not require that a property be associated with at least one individual**
  - In OWL, properties are declared as separate constructs, independent of any classes or individuals. That is, in general, OWL syntax does not require properties to be associated with any classes or to be possessed by any individual (even though it might not seem very useful to declare a property which is not used at all).¹⁰

The above-mentioned issues enable the creation of ontologies in OWL that do not comply with the fundamental ontological assumptions regarding properties. Therefore, we propose additional guidelines addressing the relationship between properties, classes and individuals in OWL-based conceptual modeling.

First, we introduce a guideline to reflect that all things possess properties:

**Guideline 3:** *Every OWL individual representing a substantial thing must possess at least one property.*

Implementation of this guideline in OWL can be done in one of two ways:

- **at an instance level** – by explicitly declaring a fact that the individual possesses a particular value for a property, or
- **at a class level** - by declaring that an individual is a member of a class which includes at least one property in its class definition.¹¹

To illustrate this guideline, consider an individual John_Doe. We associate a property (hasAge) with this individual in one of two ways. In the first case, the individual is associated directly with the property hasAge.

```xml
<owl:DatatypeProperty rdf:ID="hasAge">
</owl:DatatypeProperty>
```

¹⁰ A property in OWL can be associated with individuals either directly or through defining a class in terms of property restrictions that have to be satisfied by all instances of the class.

¹¹ Such a definition can also be inherited from another class.
In the second case John_Doe is declared as an individual of the class Person possessing the property hasAge.

The next guideline assures that every property is possessed by at least one substantial thing:

**Guideline 4:** Every OWL property intended to represent some property of substantial things must be associated with at least one OWL individual representing a substantial thing.

This guideline can be implemented in OWL using one of two ways - at an instance level or at a class level:

**at an instance level** - explicitly declare a fact about an OWL individual, i.e. a property should be used in at least one assertion about an individual possessing a specific value for the property. For example, to include hasColor as a property in a model, a fact that some (substantial) individual possesses some value for the property hasColor can be declared (e.g. `<owl:Car rdf:ID="John'sCar">`).

**at a class level** – at least one substantial OWL individual could be inferred to possess this property by virtue of being a member of a class associated with this property.

For example, the property hasVehicleNumber can be associated with the class Car using a suitable property restriction (e.g. Cardinality =1). Figure 23 describes such a situation where the individual Spider possesses the property hasVehicleNumber by becoming member of the class Car.
3.4.2 Developing conceptual models using the proposed guidelines

We demonstrate how to use the guidelines for modeling things and properties by redefining a part of wine ontology (McGuinness et al., 2004) (discussed in Section 3.2.2). The part of this ontology is shown in Figure 24. Wine and WineGrape are modeled as classes as they represent different varieties. CabernetSauvignonGrape is modeled as an instance of the class WineGrape. WineDescriptor is a class that describes wine in terms of its taste and color (therefore modeled as a union of two classes: WineTaste and WineColor).
We now apply the guidelines to this ontology. First, we create two higher-level classes: `Substantial_Thing` and `Non_Substantial_Thing` which are disjoint. The classes `WineGrape` and `Wine` are modeled as subclasses of `Substantial_Thing` because their instances are real wine grapes and real instance of wine respectively. In the original ontology, `CabernetSauvignonGrape` is modeled as an instance of `WineGrape`. Smith et al. (McGuinness et al., 2004 p. 11) justify this decision by stating that "The Grape class denotes the set of all grape varietals, and therefore any subclass of Grape should denote a subset of these varietals. Thus CabernetSauvignonGrape should be considered as an instance of Grape and not a subclass. It does not describe a subset of Grape varietals, it is a grape varietal." However, considering the definition of a substantial individual, `CabernetSauvignonGrape` cannot be considered as an individual as it does not point towards any actual or possible instance, rather it should be modeled as a subclass of `WineGrape` as it denotes a grape variety (for which physical instances exist). On the other hand, `WineDescriptor` describes wine and therefore is a type of attribute and not a class of substantial things. Therefore we model `WineDescriptor` and its subclasses under `Non-Substantial Thing`. Following guideline 3, namely, that classes must have properties the classes `WineGrape` and `Wine` are associated with at least one property each. To see why this is so, consider that wine can...
be made of different fruits. The fact that the ontology recognizes grape as the specific fruit and distinguishes among grape types, indicates that there are some characteristics that are common to all grape types (and might distinguish them from other fruits). These characteristics will be modeled as "properties of WineGrape" (for example - grapes can be fermented). Note that individual grape types might have different "manifestations" of these properties (e.g. grape size). In short, the ontological guideline "forces" a modeler to provide a reason (in the form of properties of interest) why to include a certain class of substantial things (e.g. why even include the class "WineGrape"). Without this justification, there might be no reason to include the class. The revised ontology is shown in Figure 25.

Figure 25: Revised Wine Ontology

(Substantial classes and their properties are indicated by rectangles and non-substantial classes and their properties by rounded rectangles)

12 The links between the classes Substantial_Thing and NonSubstantial_Thing indicate that these classes are disjoint.
Recall that, similar to prescribing that each substantial class must have properties, the guidelines also specify that each property is "owned" by at least one (type of) substantial things. This implies that in the formal OWL description each property must be used (directly or via a subclass of the property) by at least by one class (or instance).

Finally, we note an important distinction between modeling substantial and property classes. When dealing with substantial things, we have clear ways of distinguishing between instances and classes (as discussed above). This however is not the case for modeling Non_Substantial things such as properties. We demonstrate the difficulty of distinguishing instances from classes of properties using the WineDescriptor property example (Figure 26). The property WineColor (subclass of WineDescriptor) is viewed as a class with instances: White, Rose and Red. However, Red also has different shades and therefore we might conceive of Red as a class. Similarly individuals of the classes: WineTaste, WineBody, and WineSugar can also be considered as classes. At the end we note that the way properties of non-substantial things are organized into class and instance hierarchies reflects subjective views, rather than an objective observation that can be applied to substantial individuals.

![Figure 26: WineDescriptor hierarchy](image-url)
If we can identify the substantial individuals (and their classes) in the modeled domain, we can then identify their properties and how they can be classified. The guidelines can represent the domain facts consistently. Now assume the knowledge of a domain changes. If the ontology is modified according to the prescribed guidelines, it is unlikely that what have been identified as substantial individuals will become classes or properties, or vice versa. Any new class containing substantial instances (such as a class Fruit) will appear in the hierarchy of classes beginning with Substantial_Thing. As noted above, this, however, will not necessarily be the case for properties. Changes involving properties might cause changes of constructs used.

The above analysis suggested guidelines for modeling phenomena that can be directly mapped into OWL constructs. The main issue was how to do this consistently. We now turn to modeling phenomena for which no readily available constructs exist in OWL.

3.5 Modeling interactions in OWL

3.5.1 Difficulties in modeling some real-world phenomena in OWL

OWL has no concept to capture the behaviour or dynamics of instances or classes directly. As the concept of state does not exist in OWL, the change of state (i.e. event) cannot be represented either. A notion that is related to change of states is interaction. We now suggest how to model interactions using OWL constructs. We choose interactions for two reasons. First, when dealing with domain semantics, it conveys how different things in a domain can affect each other. Second, in the context of IS design it has been used to provide ontological meaning to the concept of relationships in entity-relationship diagrams (Wand et al., 1999) and association classes in UML class diagrams (Evermann and Wand, 2005).

When modeling interactions, we will adhere to the guidelines suggested above and will add some specific guidelines. Prior to doing this, we discuss an ontological view of interaction.

3.5.2 Interaction

According to Bunge, interaction is the ability of a thing to change the “history” (states traversed) by another thing (Bunge, 1977 p. 62). Changes to things are manifested as changes to properties, which are modeled via changes in the values of attribute functions, i.e. changes of state (Wand
The existence of interaction can be considered a mutual property of things, and conversely, the existence of a mutual property can indicate an interaction. "A mutual property that reflects interaction is termed a binding mutual property" (Bunge, 1977 p. 102). A binding mutual property implies that some changes in one thing are related to (such as precede, are accompanied by, or are followed by) changes in the other thing. For example, the mutual property "a company employs a person" implies that the existence of the (specific) company affects the state of a (specific) person (and vice versa) (Wand et al., 1999).

### 3.5.3 Guidelines for modeling interactions in OWL

Because interactions in Bunge’s ontology are directly related to mutual properties, we suggest distinguishing mutual properties from intrinsic properties. OWL provides for defining a property the value of which is another class. Such a property is called an object property. For example, an object property "EnrolledIn" links student class to university class (Figure 27):

![Figure 27: Example of OWL object property](image)

However, from an ontological point of view, the use of OWL object property to represent mutual properties has several limitations. First, in Bunge’s ontology a property cannot be a thing. Second, often, several mutual properties are related to one interaction. For example, when a company employs a person, relevant properties will be start date, job title and salary. Representing an interaction by one mutual property will not allow grouping of mutual properties and hence modeling such a situation might not be possible, leading to confusion and information

---

13 Bunge also defines non-binding properties for properties unrelated to interactions. However, henceforth we only refer to mutual properties associated with interaction and will eliminate the term “binding”.
14 We only refer to class properties, as when modeling a domain we need to include the underlying concepts, and those are represented by classes.
loss. Third, in Bunge's ontology, mutual properties can be shared by any number of things while OWL object properties can only represent binary mutual properties.\textsuperscript{15}

To overcome these problems we follow the suggestions made in Evermann and Wand (2005) and show how to employ these suggestions using OWL constructs. We formulate several guidelines, which we illustrate with a running example. Consider the situation where a person becomes a customer of a bank by opening a bank account. After a while, the customer negotiates with the bank to become a preferred customer who is entitled to hold a trading account. We first show the classes and their properties that might be included in a typical OWL representation, unguided by ontological considerations (Figure 28).

![Diagram of customer-bank domain](image)

**Figure 28: Unguided OWL representation of the customer-bank domain**

Second, in accordance with the general guidelines for modeling domain concepts, we reorganize the representation defining two disjoint upper-level classes: Substantial_Thing and Non_Substantial_Thing. Person, Customer, PreferredCustomer and Bank are identified as classes

\textsuperscript{15} In some cases, ternary or higher relationships can be replaced by a set of binary ones. However, this might lead to a loss of information conveyed by a conceptual model (Wand et al., 1999).
having substantial instances and thus are modeled as subclasses of Substantial_Thing (Figure 29).

![Class hierarchy of the bank-customer domain](image)

Figure 29: Class hierarchy of the bank-customer domain

Third, we add interactions to the domain representation. We identify two types of interaction in this example: between the customer and the bank (manifested in terms of BankAccount) and between the preferred customer and the bank (manifested as TradingBankAccount). Each of these interactions is associated with several mutual properties. For example, BankAccount involves account number, date of opening, and balance and TradingBankAccount involves trading account number, date of opening, and trading limits. To model such interactions and associated mutual properties, we propose the following guidelines:

**Guideline 5:** Sets of mutual properties of substantial things arising out of the same interaction should be represented as OWL properties of a specially defined OWL class –\textit{interaction}.

**Guideline 6:** Each interaction class represents a set of mutual properties arising out of the same interaction. Different interaction classes should be used for each type of interaction.

**Guideline 7:** An interaction class should be modeled as a conceptual class.\(^\text{16}\)

**Guideline 8:** Each interaction class must be associated with at least one mutual property.

To implement these guidelines in OWL we suggest the following rules:

\textit{Rule 2:} Interaction classes will be modeled as a subclass of Non\_Substantial\_Thing and will have at least one OWL property.

\textit{Rule 3:} Each mutual property of substantial things will be represented as an OWL property of an interaction class.

---

\(^{16}\) Because it does not represent substantial things but rather “bundles” of mutual properties of substantial things
In our example, we will create two OWL classes to represent the two interactions: BankAccount and TradingBankAccount (guidelines 5 and 6) each a subclasses of Non_Substantial_Thing.

For practical reasons, we suggest the use of a prefix (e.g. mp_) in naming a mutual property (associated with an interaction class) to differentiate it from other types of properties. The rules require that we define OWL properties to represent these mutual properties and associate them with the respective interaction classes. For the property bank account number, we can create an OWL property mp_AccountNo (whose domain is the interaction class BankAccount).

Recall, a mutual property is actually a property of each of the interacting individuals. Thus, we need a way to attribute the mutual properties to the involved individuals. This is the purpose of the next guideline.

**Guideline 9:** Each interaction where a substantial thing is involved should be modeled by a property of the class whose value is an instance of the interaction class.

We implement this guideline with the following rule:

**Rule 4:** Each class of substantial things participating in an interaction will have an OWL object property reflecting this interaction.

To improve model understandability, we suggest that the interaction class should explicitly refer to Substantial_Thing classes that participate in the interaction.

**Rule 5:** Each interaction class should have a property for each involved class whose value is an instance of the involved class.

To demonstrate the rules, a simple example is given in Figure 30 where the Customer and Bank instances are shown explicitly to participate in BankAccount interaction by including the OWL object property “ParticipatesInBankAccount” for the customer class.

---

17 Recall, a property whose value is an instance of a class, is termed an object property in OWL.
In this example, we show how to indicate explicitly that the Customer and Bank instances participate in BankAccount interactions, the OWL object property "ParticipatesInBankAccount" is defined for the customer class.

For practical reasons, we suggest using a prefix (e.g. ParticipatesIn_) to name an object property indicating participation in a particular interaction and a prefix (e.g. Involves_) in the object property name to show that this property is used to link the interaction class to the involved Substantial_Thing classes.

We now turn to the full example to demonstrate the application of the guidelines. We add the two interactions to the model of Figure 30. The complete model is shown in Figure 31.
Recall (from Section 3.5) that an interaction occurs when one thing affects changes in another thing. A key advantage of modeling interactions explicitly by following the suggested guidelines is that it enables tracking such changes. The difference between a person and a customer is that the customer participates in the BankAccount interaction while the person does not. Thus, a Customer possesses all properties of Person and the additional ones associated with the interaction involved in opening a bank account. Similarly a PreferredCustomer participates in the interaction TradingBankAccount, and acquires all associated properties in addition to those of a regular customer. Thus, the guidelines enable us to model the changes that can occur as a result of interactions.
We end the above analysis with some additional notes:

(1) The guidelines suggest a method to check the models. For each interaction class there must exist at least two object properties linking individuals of the Substantial_Thing class (or one of its subclasses) to the interaction class.

(2) As discussed above, when an instance of a class participates in an interaction, it acquires additional properties. This acquisition effectively makes it a member of a subclass of the original class (for example – a Preferred Customer becomes a subclass of Customer).

(3) The guidelines imply that all members of a class involved in an interaction must possess the relevant mutual properties. This is in line with the modeling rules proposed in (Wand et al., 1999) which prescribe that when a class instance is involved in an interaction it becomes a member of a subclass where interaction is mandatory.

3.5.4 A Sample Case

To further illustrate how the guidelines can be implemented, we use the auction ontology (www.schemaweb.info, 2003). An auction is a competition-based method of allocating scarce resources and involves three types of participants: seller, auction house and buyer. The two actors creating the auction are the seller who owns the goods (the Auction Item) and the auction house person who runs the auction on the seller’s behalf. Two important characteristics of an auction are closing time and reserved price. A bidder is an agent who participates in the auction and submits the bid which has an expiry time. The resource (such as a deposit) reserved for the bid by the bidder is referred to as reserveResource. An auction is initiated when a seller places the items intended for sale with the auction house, where the auction takes place. Bidders place bids on an item for sale. The item is sold to the highest bidder. Figure 32 shows the original ontology in which participant appears instead of buyer. Notice that Auction appears as a class, while Seller and auctionHouse are not shown directly, but are shown as properties of the class Auction. Similarly, bidder appears as a property of the class Bidding.
While syntactically correct, we claim that this is not a clear representation of the domain, as physical things (seller, auction house, bidder) do not shown as classes, while processes and actions (auction and bidding) are considered classes. We now reorganize the domain information following the guidelines. First, classes of substantial things are identified. These are: AuctionItem, ItemSold, Bidder, Buyer, and AuctionHouse. Next, we identify the relevant interactions in the domain. These are: auctionedAt (interaction between the AuctionItem, Seller and the AuctionHouse) bidding (interaction among Bidder, AuctionHouse and the AuctionItem) and WinningBid (interaction among Buyer, itemSold and AuctionHouse). According to the interaction modeling guidelines, we model these three interactions as subclasses of Non Substantial Things. All classes identified are shown in Figure 33.
The next step involved identifying the mutual properties related to each of these interactions. Finally, each interaction class is related to the classes of substantial things involved in the interaction. The resulting ontology is depicted in Figure 34.

We briefly note some differences between the original ontology (from (www.schemaweb.info, 2003), shown in Figure 32). In the original representation, auction is treated as a class. However, in the new representation (Figure 34), auction is not shown explicitly, but rather is represented by the whole model. On the other hand, the interactions comprising the auction are shown. Note that closing time and reserved price which are characteristics of Auction are now mutual properties of the auction house and the auctioned item. The actual process of auctioning and its effects are implied in the modified representation via the interactions among the participants. For example, in the modified representation it is clear that a bidder becomes a buyer via participating in the WinningBid interaction and thus acquired the property- ParticipatesInWinningBid (which is not a property of the Bidder class). Since a buyer is modeled as a subclass of bidder, it can be inferred that the bidding interaction (involving Bidder) precedes the interaction WinningBid (involving Buyer). The distinction between AuctionItem and ItemSold is also made explicit in this ontology. As the result of the interaction-WinningBid, the ItemSold class acquires an additional property- participatesInWinningBid to that of the AuctionItem class.
3.6 Conclusion

Conceptual models are important tools used to formalize domain understanding in support of information systems development. Such models should be precise, yet understandable to stakeholders. Independent of conceptual modeling, ontology description languages have evolved for the purpose of sharing computer-readable representations of terminologies. In this paper we set out to suggest how such languages can be used to develop conceptual models. We first observed that the key to this use of ontology definition languages is suggesting clear mapping
guidelines from general domain concepts to the language constructs. Beyond that, domain phenomena exist for which no direct mapping into language constructs can be found. Hence, we add to the mapping guidelines suggestions on how to use extant language constructs to represent such phenomena.

Our proposed guidelines are derived from Bunge's ontology. This ontology uses concepts derived from science, and thus might be perceived as a "realist" one, while our objective is to model human perceptions of domains. We contend, however, that this should not preclude adapting the ontological concepts as they can still be used to convey perceptions of domains. In this paper, we have demonstrated, using OWL constructs, how certain domain phenomena can be made clearer when following the proposed guidelines. However, the question remains as to how effective a set of specific ontological concepts is in conveying domain semantics as perceived by humans. As a given ontology is effectively a set of beliefs, we claim this can only be addressed by empirical work.

This research can be extended in several ways. First, we have used Bunge's ontology as a philosophical base. However, other choices are possible. Further research can explore the use of other philosophical ontologies that have been deployed in the context of information systems. Second, other domain phenomena for which there is no direct representation in OWL, exist. Notably, these include composites, laws, states and events. Future research can explore mechanisms to model additional constructs in OWL (in particular, the notion of a composite is an obvious candidate for such analysis). Third, the outcomes of our analysis do suggest some ideas for possible extensions and modifications to OWL (such as developing new constructs to model interaction). This, in turn, can point out at some potentially useful functionality for ontology development tools in OWL which facilitate the development of better domain concepts. Finally, Kishore et al. (2004) mention the need of research on ontology development methodologies. Our guidelines can be useful for this purpose.
References


Chapter 4

Ontological Support for Knowledge Identification: A Theory and an Empirical Study\(^{18}\)

4.1 Introduction

Managing knowledge in organizations is considered very important for business. Industry reports indicate that by 2005, most leading organizations utilized knowledge management (KM) practices to support critical business processes, and many had implemented enterprise-wide KM programs (Gartner, 2005a). To support such programs, organizations often deploy knowledge management systems (KMS) (Alavi and Leidner, 2001). KMS are IT systems that provide functions to support the management of explicit and tacit knowledge (Maier, 2002). Examples of functions supported by KMS are the organization, search and retrieval of knowledge in explicit (i.e. documented) forms and the identification of experts who possess implicit knowledge (Wakefield, 2005).

To use a KMS effectively, a knowledge seeker should be able to identify the required knowledge. However, users often seek knowledge in areas not completely familiar to them and therefore might not be clear about which knowledge to search for. Hence, an important issue in KMS use is identifying the required knowledge. Thus, it is desirable that a KMS supports users in identifying their knowledge needs, prior to accessing the knowledge itself. To distinguish this step in knowledge search we term it knowledge identification.

Much of the research about KMS focuses on design issues such as identifying knowledge management (KM) processes to be supported by KMS (Gallupe, 2001). In particular, significant research has been done on KMS functionality needed to support knowledge storage and transfer (Alavi and Leidner, 2001). However, not much research on how individual users actually use KMS exists. Specifically, knowledge identification in the context of KMS use appears to be under-studied. For example, Gallupe (2001, p. 71) mentions “[v]ery little research has been

----

\(^{18}\) A version of this chapter is submitted to Information Systems Research for publication. Bera, P., Wand, Y. and
conducted on tools and technologies to support practices focusing on new and unique knowledge problems.” Similarly, Hahn and Subramani (2000) mention that the KM literature has largely focused on general conceptual principles of KM and KMS without much guidance as to how to conceptualize the requirements of a KMS. Rouse (2002, p. 283) mentions "lack of understanding of decision makers' needs to know - and provision of support systems accordingly - is precisely the missing link in the success of many information and knowledge management systems."

There has been limited research on methods for knowledge identification. Kwan and Cheung (2006) mention that potential knowledge recipients first need to identify the gaps between existing knowledge and target knowledge needed to accomplish a task. They suggest that this gap can be reduced using brainstorming tools, search tools, knowledge repositories and knowledge maps. However, not much research exists on how users should use such tools in knowledge identification. In the knowledge engineering (KE) domain, knowledge acquisition techniques are popular. These techniques aim at acquiring knowledge from an expert and transferring the acquired knowledge into a computerized form (such as an expert system) (Preece et al., 2001). However, such techniques are intended to obtain knowledge from experts, where the required knowledge is already identified, not to help identify which knowledge is needed.

We conclude that proper approaches to support knowledge identification are required. Following this, this paper suggests how knowledge identification can be supported in KMS by using formal ontologies, and describes an empirical test of the suggestions. Specifically, the paper proposes the use of visual formal ontologies guided by some ontological principles. A formal ontology is a set of concepts and relationships among these concepts intended to describe a particular domain (Gomez-Perez et al., 2004). Following this, the paper describes an experiment intended to test the effect of using such guided ontologies on effectiveness of knowledge identification.

---


19 Note, there are two related, but different uses of the word “ontology”. In philosophy, an ontology can be defined as “That branch of philosophy which deals with the order and structure of reality in the broadest sense possible” (Angeles, 1981). In the context of computerized system, a formal ontology is a computer-readable formal definition of a set of concepts (Gomez-Perez et al., 2004).
The rest of the paper is organized as follows. In section 4.2, we discuss knowledge identification, referring to the information search literature and suggest the use of formal ontologies to support this task. In section 4.3, we discuss the use of formal ontologies to support knowledge identification. In Section 4.4, we suggest how formal ontologies can be constructed to support knowledge identification. Section 4.5 poses hypotheses about the effective use of formal ontologies and describes an experiment to test the hypotheses. Section 4.6 describes the results and discusses their implications. Finally, Section 4.7 concludes by discussing the contributions of the research and its limitations and suggests future research directions.

4.2 Knowledge Identification

4.2.1 Defining knowledge

To study knowledge identification we first need to adopt a clear concept of knowledge in the KMS context. The KM literature provides divergent definitions of knowledge, from "justified beliefs" (Nonaka and Takeuchi, 1995) to "experience and values" (Davenport and Prusak, 1998). However, no generally accepted definition is available (Spender, 2003). We take a position that the purpose of KMS is to support organizational actors in performing their tasks by providing access to representations of and references to knowledge. Hence our approach draws on the pragmatic philosophy which views knowledge in terms of actions (Blosch, 2001) and on an Artificial Intelligence (AI) view of knowledge representation as proposed by Newell (1982). Newell relates knowledge to agent, action and goals. His view of knowledge is widely recognized in the AI community (Musen, 2004).

Newell addresses two levels: the knowledge level that deals with the nature of knowledge, and the symbol level that deals with the representation of knowledge. Each level comprises components, laws of composition that enable components to be assembled into systems, and laws of behavior (Newell, 1982). The knowledge level attributes to an intelligent system (an agent): knowledge, goals, and actions. These are, in turn, represented explicitly in a physical symbol system (Newell, 1982). The agent processes its knowledge to determine what actions to

20 We refer here to the categorization of KMS as being of two types: repositories and networks (Alavi and Leidner, 2001).
take in order to attain its goals. Musen (2004, p. 77) mentions: "Knowledge does more than account for what exists in the world, it directly links goals to actions. Knowledge lets an agent enact procedures to attain its goals."

We will use the concepts of agents, their actions and goals to operationalize knowledge identification. We therefore briefly describe the agent concept as addressed in the AI literature. An agent can be defined as an entity that acts upon the environment it inhabits (Wooldridge, 2000). Thus agents can bring about changes in the environment (Castelfranchi, 1998). An agent’s environment contains a collection of objects (agents and resources) that agents are able to (at least partially) perceive and manipulate (Logie et al., 2003). Agents and their environment have states (Brafman and Tennenholtz, 1997). The goals of an agent reflect the desires the agent intends to fulfill (Logie et al., 2003; Luck and Inverno, 2001; OMG, 2000). These goals are a subset of the agent’s and its environment possible states. An agent possesses a repertoire of possible actions it can perform to modify the current state (of the environment) (Rudowsky, 2004; Wooldridge, 2000). Since an agent’s behavior is aimed at producing some result, we are referring to goal-oriented agents (Conte and Castelfranchi, 1995).

The agent-oriented approach towards knowledge is quite common in KM literature. For example, Elst et al. (2003) mention that agents can be seen as a natural metaphor to model KM environments. Similarly, Nabeth et al. (2003) mention that a KM environment contains a set of knowledge resources, mechanisms, and agents. In this environment, the agents access the resources, participate in the creation of new knowledge resources, interact and exchange knowledge.

4.2.2 Defining knowledge identification

We conceive of organizational actors as agents who obtain knowledge to reach goals. Based on the above, we suggest that an agent uses a KMS to solve problems defined in terms of current and desired (goal) states. The link between knowledge, problem solving and goal states has been described, for example, by Gallupe (2001, p. 67) who states: "KMS are used to solve problems, "We define problems in the context as “desired states”." Accordingly, he suggests that KMS should support problem related practices including: identifying the problems that can be solved, storing the knowledge to solve the problem, and, transferring the stored knowledge to others.
However, problems for which KMS support is needed are often not apparent or simple and have been even referred to as “wicked problems” (Gallupe, 2001). Indeed, it can be claimed that the essence of knowledge management is dealing with such problems. In such cases, a precise a-priori description of how to execute a task or solve the problem does not exist (Elst et al., 2003). Moreover, the KMS user cannot always be expected to know which knowledge is required for solving the problem. This brings about the importance of identifying the knowledge needed. For example, KPMG (2003) mentions knowledge requirements identification as a process of discovering what knowledge is needed and how to obtain and maintain it. Kwan and Cheung (2006) mention that potential knowledge recipients first need to identify the gaps between existing knowledge and target knowledge needed to accomplish a task. This issue has been recognized in the information science literature. Meho and Tibbo (2003) observe that information searching activities do not necessarily always start with or lead to direct sources of information. They suggest the usefulness of information retrieval systems can be improved by including features in these systems that directly support activities defined in information search process models. Drawing a parallel to knowledge search, we suggest that a KMS user needs to identify what knowledge she needs before searching for it, otherwise she will not know what to search for, where to search for it, or how to assess what she finds.

Accordingly, we suggest that integrating a feature to support knowledge identification in KMS may help users to use KMS effectively.

At this point, we note the distinction between problem solving and knowledge identification. Problem solving refers to a task which an actor needs to perform. To do this, the actor requires knowledge to help specify the actions that need to be taken (and how to perform them). To obtain the required knowledge, the actor needs to identify what it is. This is knowledge identification. For example, assume a travel agent needs to cancel a reservation. The problem solving task is how to do this. Knowledge identification is to determine the “right” questions that need to be posed in order to identify the necessary actions and how to perform them.

A KMS is used specifically to provide the knowledge that will help an agent to solve the problem. However, the agent might not even know what knowledge to search for. Thus, the
agent needs first to identify which knowledge is needed. Recalling our view of knowledge in terms of agents, actions, states, and goals, we now define *knowledge identification* as the task of asking the questions necessary to find the actions required to change the current state of affairs to a goal state. We now turn to the question how the required knowledge can be identified in the KMS context.

4.3 Supporting knowledge identification

4.3.1 The general approach

To conceptualize knowledge identification tasks in KMS, we refer to the information science literature dealing with information search. Information seeking is defined as a process in which humans purposefully engage in order to change their state of knowledge (Marchionini, 1995). Borgatti and Cross (2003) mention that information seeking and knowledge seeking are related constructs. The information search literature often highlights the need to identify the information required. Specifically, this literature indicates that a user recognizes an information need and goes to the retrieval system with a request based on that need (Ellis, 1993). Accordingly, a distinction is made between retrieval systems that help identifying required information and retrieval systems that actually provide information (Cole and Leide, 2003). This distinction is apparent in information search process models (e.g. (Ellis, 1993), (Kuhlthau, 1991, 1993)). Kuhlthau (1991; 1993) found that there are six clearly defined activities in an information search process: task initiation, topic selection, prefocus exploration, focus formulation, information collection, and search closure. Kuhlthau's information search model has been tested empirically in several studies (including search for digital information) (Spink et al., 2002).

An important stage in these process models is *domain familiarization*. For example, in Kuhlthau's process model (1991), the third stage of the information search process is to explore an unfamiliar topic to gain a personal understanding of it. Similarly, Pennanen and Vakkari (2003) mention that user's insufficient conceptual understanding of the domain can have a major impact in information search tasks. They claim that when users have a richer and more structured conceptual representation of what information they need, they can formulate more focused and complete queries. Cole and Leide (2003) recognize that users who are novices to a
domain do not know the structure of the topic area. Thus, they suggest that a classification scheme (a concept hierarchy) that represents the structure of a topic area can help users identify their information needs. They further suggest that relating this structure with the user’s mental model can help the user define his/her topic more clearly and better identify his/her information need.

Drawing a parallel between information search and knowledge search, we suggest that to perform knowledge identification, users need to be familiarized with the domain concepts. This familiarization can be especially useful if users have inadequate domain understanding.

4.3.2 Using formal ontologies

Following the suggestion that domain familiarization will help users perform knowledge identification we propose the use of formal ontologies for this purpose. Formal ontologies are used in applications that need to share subject-specific (domain) information (Gomez-Perez et al., 2004). Several roles of formal ontologies in KMS are mentioned in the literature. First, ontologies are used to represent knowledge (Maedche et al. 2003), second, ontologies can be used to facilitate designing of mechanisms to enable retrieval of knowledge and identification sources of knowledge (O'Leary, 1998). These two roles are both design-related. Our interest is in supporting the use of a KMS, specifically, the knowledge identification step preceding knowledge search and retrieval. In order to know the meaning of knowledge available via a KMS, users need to first understand the domain about which they seek knowledge. This is in line with the observation of Pretz at al. (2003) that prior to engaging in problem solving, people need to form a correct mental representation of the problem. We are interested in the use of formal ontologies to help users understand the domain about which they intend to search knowledge. Such understanding will help users identify the knowledge required to perform a given task. In the context of using KMS, Rao and Osei-Bryson (2006) suggest that KMS provide a facility for searching or querying an ontology by the end users “in order for them to get a clear understanding of the domain.”

We propose the use of visual ontological representations to facilitate knowledge identification. This suggestion is in line with the suggestion of using diagrams to help people
understand and solve problems (Novak and Bulbo, 1990). We now discuss how such a suggestion can be implemented in practice using available ontology description languages.

### 4.3.3 Using visual descriptions of formal ontologies

The importance of using formal ontologies to support sharing of information resources is widely recognized (Gomez-Perez et al., 2004). Several languages are now used to publish and share formal ontologies on the web (McGuinness et al., 2004). A commonly used ontology description language is OWL (created by the W3C ontology working group) (McGuinness et al., 2004). OWL is considered one of the key Semantic Web technologies that provide a framework for data sharing and reuse on the Web (Gomez-Perez et al., 2004). OWL enables the definition of classes, individuals, properties of classes and individuals, and assertions about properties (for a brief description of OWL see Appendix B).

Several tools have been developed to represent ontologies visually, such as the Protégé-OWL plugin (Protege, 2003) and KAON (Motik, 2002). Visual ontologies are often used in knowledge portals to provide a structure of the knowledge base (Maedche et al., 2003). For example, KAON is a web-based environment that allows users to browse, search, and navigate the knowledge base portals (Staab et al. 2001). O'Leary (1998, p. 38) claims that visual representations “would allow a user to visually follow a concept to its nearest neighbors or analyze the overall space for interesting related or unrelated concepts.”

To demonstrate how a visual ontology diagram can facilitate knowledge identification, we provide an example. Assume that a person becomes a bank customer by opening an account. Figure 35 provides a visual representation of a set of OWL statements describing this situation (drawn using ezOWL visualization tool in Protégé (2003)). The ontology includes bank, person, customer, and their interrelationships. It shows that a customer can become a preferred customer who holds a trading account. In the context of a bank’s KMS, an example of a knowledge identification task would be: “what knowledge is required to change the state of a customer to a preferred customer?”
Examining the diagram, we notice that it shows the links among the different concepts. This is typical to such diagrams. However, the diagram does not specifically exhibit the activities that may lead to changing the status of the customer, and hence might not provide sufficient guidance to the user seeking to identify relevant knowledge. In the next section we suggest how such information can be explicitly represented in a formal ontology, by applying principles taken from a philosophically-based ontology.

4.4 Guiding the Use of Formal Ontologies

Above we suggested the use of ontology diagrams to support knowledge identification. However, we observe several difficulties related to this suggestion. First, ontology description languages are designed for software applications rather than to support domain understanding. Therefore, they might not be able to convey domain information properly, and might lack the capacity to reflect some important domain phenomena, especially those related to domain dynamics (Bera and Wand, 2004). Second, knowledge seekers cannot be expected to understand formalized ontologies, thus visual representations have to be made "rich" enough to convey the relevant information. Third, no rules exist on how to develop and present formal ontologies (Gomez-Perez et al., 2004).
To alleviate these difficulties, we propose to add semantics to formal ontological descriptions by using guidelines derived from ontological (i.e. philosophy-based) theories. Specifically, we suggest that ontology diagrams developed by following these guidelines can help KMS users understand the domain and thereby improve knowledge identification performance. Accordingly, we propose to test empirically the question: will a visual representation of a domain ontology that complies with ontological guidelines provide better support for knowledge identification than a visual representation of a domain ontology that does not comply with ontological guidelines? We now turn to the question of how to improve the expressiveness of a formal ontology description of a domain.

4.4.1 Improving the expressiveness of OWL ontologies

We now suggest how to create more expressive ontologies in OWL. To balance between added expressiveness and complexity, and to enable controlled testing we focus on one specific representational issue that we consider especially relevant to knowledge identification. Specifically, we suggest that ontologies developed in OWL be modified to improve the way that they represent “interactions” among things in a domain. The notion of interaction is related to the change of state of an entity. To see why we chose the interaction concept recall that above (Section 4.2.2) we defined knowledge identification as the task of asking the questions necessary to find the actions required to change the current state of affairs of an entity to a goal state. Next we define the concept of interaction and how it can be modeled in OWL.

4.4.1.1 The Interaction Concept

Since OWL does not provide a direct representation of interactions we seek guidance in an ontological theory. We use Bunge’s (1977) ontological theory, which offers a detailed description of the notion of interaction.21

The guidelines are based on a mapping between the Bunge’s ontology constructs and the OWL language constructs. Such a mapping allows the assignment of ontological semantics to

21 We note that other ontological approaches could be used (Chisholm, 1996; Guarino and Welty, 2002; Sowa, 2000). However, our interest is not in comparing ontologies, but in showing that their use can be advantageous. We chose Bunge’s ontology because of its high level of formality and because it explicitly defines actions and interactions.
OWL language constructs. Similar mappings have been done using languages such as ER (Wand et al., 1999), NIAM (Weber and Zhang, 1996) and UML (Evermann and Wand, 2005). These prior mappings resulted in the generation of several theoretical predictions. Moreover, several of these predictions have been empirically examined and corroborated ((Bodart et al., Weber, 2001; Burton-Jones and Meso, 2006; Gemino and Wand, 2005). Accordingly, we believe that when OWL ontologies are developed in accordance with the guidelines developed in this study, they will represent the concept of interaction more effectively.

Bunge’s ontology specifies certain relationships and constraints related to things, properties and classes. In particular: (a) All things have properties (which exist whether humans are aware of them or not), and (b) properties are always attached to things (“every property is possessed by some individual or other; there are no properties that fail to be paired to any individuals”, (Bunge 1977, p.62)). In Bunge’s ontology, a property that is inherently a property of a thing is called an intrinsic property (e.g., age) and a property that is meaningful only in the context of two or more things is called a mutual property (e.g. “salary of an employee,” which is mutual to an employee and an organization). Properties are modeled as state functions, the values of which form the state of the thing. In Bunge’s ontology, an interaction occurs when one thing affects the states traversed by another thing and is manifested via mutual properties. When such interactions occur, they affect the state variables in each thing, and thus result in a mutual property between the two things. For example, by interacting with a company (and thus obtaining the mutual property Employee_Salary), the state definition of a person is changed to that of an employee. In the following, when there is no room for ambiguity, we will use the term “property” rather than the term “state function.”

4.4.1.2 Modeling interaction in OWL

Because interactions in Bunge’s ontology are directly related to mutual properties, we suggest that to model interactions in ontologies developed in OWL, it would be beneficial to have a way to distinguish mutual properties from intrinsic properties. OWL provides for defining a property
the value of which is another class.\textsuperscript{22} Such a property is called an object property.\textsuperscript{23} For example, an object property "EnrolledIn" links student class to university class (OWL syntax shown below):

\begin{verbatim}
<owl:Class rdf:ID="Student"/>
<owl:Class rdf:ID="University"/>
<owl:ObjectProperty rdf:ID="enrolledIn">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#University"/>
</owl:ObjectProperty>
\end{verbatim}

However, from an ontological point of view, the use of the OWL object property construct to represent mutual properties has several limitations. First, object properties in OWL do not directly provide for the Bunge-type mutual properties, as in Bunge's ontology a property cannot be a thing. Second, often, several mutual properties are related to one interaction. For example, when a company employs a person, relevant properties will be start date, job title and salary. Representing an interaction by one mutual property will not allow grouping of mutual properties and hence modeling such a situation might not be possible, leading to confusion and information loss. Third, in Bunge's ontology, mutual properties can be shared by any number of things while using OWL object properties can only represent binary mutual properties.\textsuperscript{24}

Following Bunge's ontology, the basic tenet of these guidelines is that mutual properties should be distinguished from intrinsic properties, and thus be modeled separately. In the context of modeling interactions in UML, Evermann and Wand (2005) propose that bundles of mutual properties should be modeled separately under a class whose name reflects the interaction that results in the generation of these mutual properties. Accordingly, we make a distinction between an interacting class (representing entities in the domain) and an interaction class (Table 1). Once these two types of classes are identified along with their properties, they should then be linked properly to trace the relationships between interacting classes and interaction classes. By

\begin{footnotesize}
\begin{itemize}
  \item\textsuperscript{22} We only refer to class properties, as when modeling a domain we need to include the underlying concepts, and those are represented by classes.
  \item\textsuperscript{23} In OWL an object property is a property whose "value" is another object. This can be shown graphically as a link between two classes.
  \item\textsuperscript{24} Although in some cases, ternary or higher relationships can be replaced by a set of binary ones, the use of such binary representations instead of a ternary one may lead to a loss of information conveyed by a model (Wand et al., 1999).
\end{itemize}
\end{footnotesize}
modeling bundles of mutual properties separately under the interaction class, the guidelines will enable development of OWL diagrams that more explicitly represent interactions between things in the domain. Based on the above, we propose a set of guidelines to model interactions in OWL (Table 8).

| Guideline 1: Identify classes the instances of which interact with instances of other classes. Identify these interactions and model them as interaction classes. Classes whose instances interact are modeled as interacting classes. |
| Guideline 2: Identify the properties of the interaction classes. The properties of the interaction classes are the mutual properties of the interacting classes related to the specific type of interaction. |
| Guideline 3: Indicate explicitly which interacting classes are involved in interactions. This can be done using object properties that link the classes that are involved in interaction with the interaction classes. A prefix (e.g. involves) can be used to indicate object properties of interaction classes that represent mutual properties of the related interacting classes. |
| Guideline 4: Indicate explicitly in which specific interactions the interacting classes participate. This can be done using object properties that link the classes that are involved in interaction with the interaction classes. A prefix (e.g. participatesIn) can be used to indicate object properties in the interacting classes which refer to interaction. |

Table 8: Guidelines derived from ontological theory for modeling interactions in OWL

To rigorously follow these guidelines, we propose some specific rules that can guide both the formal description and its visual representation (see Appendix C).

The situation described by Figure 35 can be modeled following the guidelines. We identify two types of interaction in this example: between the customer and the bank (manifested in terms of CustomerAccount) and between the preferred customer and the bank (manifested as CustomerTradingAccount). Each of these interactions is associated with several mutual properties. For example, CustomerAccount to (account number, date of opening, balance) and CustomerTradingAccount to (trading account number, date of opening). To explicitly indicate that the Customer and Bank instances participate in CustomerAccount interactions, the OWL object property "participatesInCustomerAccount" is defined. Similarly the object property participatesInCustomerTradingAccount is defined. The resulting ontology is shown in Figure 36.

25 These guidelines are derived from the guidelines presented in Chapter 3 (3.5.3)
Figure 36: Visual representation of the bank ontology developed using the guidelines

4.5 The empirical study

4.5.1 Research hypotheses

A key advantage of modeling interactions explicitly is the ability to track state changes of things. For example, in Figure 36 the difference between a person and a customer is that the customer participates in the BankAccount interaction while the person does not. Thus, a Customer possesses all properties of Person and the additional ones associated with the interaction involved in opening a bank account. Similarly a PreferredCustomer participates in the interaction TradingBankAccount, and acquires all associated properties in addition to those of a regular customer. Thus, the guidelines enable us to model the changes that can occur to things in a domain as a result of interactions. We therefore claim that the proposed modeling guidelines enable the representation of the consequences of the domain dynamics related to interactions. In the following we term ontologies developed in OWL using the guidelines informed ontologies and those developed without applying the guidelines uninformed ontologies.

Recall, in section 4.2.2 we defined knowledge identification as determining the “right” questions that need to be posed in order to identify the necessary actions and how to perform
them. Also recall from section 4.4.1.1, that an interaction occurs when one thing affects the states traversed by another thing and is manifested via mutual properties. A state definition changes interactions by obtaining mutual properties. Explicit representation of such change can be modeled in OWL by following the interaction modeling guidelines. Thus, we expect a user who views a diagram in which interactions are modeled explicitly, will better understand how changes can happen in the domain than if a user views a diagram without explicit representation of interactions. As knowledge identification tasks involve finding out what needs to be done to change the state of one or more things in the domain. Therefore, we posit that informed ontology diagrams will provide better support to users performing knowledge identification than uninformed ones.

The difference between implicit and explicit representations of interaction may appear subtle, but its effect on performance of knowledge identification task may be significant. Mayer (1983) refers to this issue by pointing out that subtle differences in the way a problem is presented can lead to considerable differences in how a subject assimilates the problem. Such differences in turn can impact problem solving performance. In our view, diagrams that represent interactions explicitly, are better representations not because they contain more information, but because they represent the same information in a better way by highlighting information (related to the state change of entities) that is useful for knowledge identification. Accordingly, we hypothesize that a user given informed ontology diagrams will be better able to identify the required knowledge items for accomplishing a task than a user given an uninformed ontology:

\textit{H1: Subjects conducting knowledge identification using informed ontology diagrams will perform better than subjects using uninformed ontology diagrams.}

To know if ontologically informed diagrams will provide better support for knowledge identification tasks than uninformed diagrams, it is important to also examine users' perceptions. This is because users will not necessarily adopt a technique if they do not perceive its advantages (Rogers, 1995). Thus, drawing on past research ((Gemino, 1998; Weber, 1997)), we test whether users perceive ontologically informed diagrams to provide them with \textit{more} understanding of a domain and whether they perceive them to be \textit{easier} to understand when compared to uninformed diagrams. Consistent with H1, we believe that ontologically informed diagrams will
help users gain more understanding about a domain because such diagrams provide them with more explicit information about the interactions that occur in that domain. Because the information about these interactions is explicit, rather than implicit, we also expect that ontologically informed diagrams will be easier to understand because it will take less effort to understand the interactions in the domain. Thus, we pose the following hypotheses:

**H2**: Subjects conducting knowledge identification tasks using informed ontology diagrams will have a higher perceived understanding of the domain depicted in the diagrams than subjects using uninformed ontology diagrams.

**H3**: Subjects conducting knowledge identification tasks using informed ontology diagrams will have a higher perceived ease of understanding of the domain depicted in the ontology diagrams than subjects using uninformed ontology diagrams.

To summarize, our independent variable is the treatment: informed versus uninformed ontology. The dependent variables are: performance on knowledge identification tasks (KNOWID), perceived understanding (PU) and perceived ease of understanding (PEOU).

### 4.5.2 Operationalizing the variables

#### 4.5.2.1 Operationalizing knowledge identification

In the information search literature, empirical studies have been conducted to identify information seeking behavior. For example, Zach (2004) asked subjects to identify what information they need to perform a task. In the present study, a similar strategy (i.e. asking users to identify what knowledge they require to perform a given task) may not work as the term “knowledge” might be interpreted differently by different people. Thus we need to operationalize the knowledge identification question without using the word “knowledge.”

To determine how to operationalize knowledge identification, we refer to the knowledge engineering (KE) literature. Analyzing several knowledge acquisition techniques in KE, Shadbolt and Milton (1999) summarize a general method for acquiring knowledge from an expert during “knowledge acquisition.” In this method, the first two steps are: (a) conducting an initial interview with the expert to scope what knowledge should be acquired and gain understanding of key domain terminologies, (b) transcribing this document to produce a set of
questions that cover the essential issues across the domain and that serve the goals of the knowledge-acquisition exercise. We follow this idea of asking questions about essential issues rather than directly asking about the knowledge, except that we propose to use such questions when asking knowledge seekers, not to knowledge "owners."

Again, recall from section 4.2.2 that the knowledge identification task is to determine the "right" questions that need to be posed to the KMS in order to identify the necessary actions and how to perform them. Referring to this definition and the idea of asking questions about a task (rather than about the knowledge) we used the following strategy.

A task can be described in terms of initial and desired (goal) states of entities in the problem domain. Knowledge identification is operationalized as a set of questions that someone might ask when developing a procedure (where a procedure refers to a set of actions performed to accomplish a goal).

To demonstrate, consider a student who needs to renew an expired library membership. In this case, the state of an entity (student) has to be changed from "without membership" to "with membership." Knowledge is the ability to select actions to change the student's state. To operationalize knowledge identification, we can ask questions such as: "You are asked to develop a procedure that a librarian would follow to renew a student's library membership. Please specify the questions you will ask to find out how to do this task." An example answer would be "How to locate a student record." This is a useful piece of knowledge when developing a procedure because it is relevant for all students and because students' records can be located in several places (e.g., physical files or databases). Note that this response is related to the selection of actions.

We emphasize again that we do not require subjects to create procedures but rather require them to come up with questions for pieces of knowledge that would help form procedures.

Although it would be ideal to test our research question in the field, the use of ontology diagrams to support knowledge identification is so new that it would be difficult to find an organization that would serve as a case site. Thus, we traded off the external validity of a case study for the internal validity of a laboratory experiment. To design the experiment, we followed
techniques applied in studies comparing effectiveness of conceptual modeling techniques. In such studies, researchers often ask users to perform “inferential” problem-solving tasks in which subjects answer questions about the domain by drawing inferences from diagrams (Burton-Jones and Meso, 2002; Gemino and Wand, 2005). Likewise, when subjects perform knowledge identification tasks in our study, they are required to identify knowledge (needed to choose actions) using the ontology representations. To develop these questions, subjects must come up with ideas by drawing inferences from the diagrams.

A pre-pilot study with 5 participants was conducted to understand how subjects respond to knowledge identification questions. We found that several subjects had difficulty understanding the nature of knowledge identification. Specifically, some subject responses did not include questions that were specific enough to form procedures. Therefore, to enable subjects to provide useable responses, a training exercise was conducted at the beginning of the experiment. In the training, subjects were introduced to OWL concepts and then asked to perform a practice knowledge identification task. The answers that subjects provided were then discussed with them and if the answers contained responses that did not help to form procedures, this was pointed out to the subjects. One criticism of this approach might be that the training exercise may reduce the generalizability of the experiment by reducing the range of ways that individuals might interpret the nature of a knowledge identification task. Although we recognize this concern, training was important in our view because it ensured that subjects understood the experimental tasks thus contributing to increased internal validity. The training manual is mentioned in Appendix D.

4.5.2.2 Perceived variables
To develop measures of perceived understanding and perceived ease of understanding, we again drew upon studies on conceptual modeling. We observed that many of these studies report that users do not perceive significant differences between representation methods even if results for actual understanding show significant effects. Past studies have suggested several reasons for not obtaining significant findings, such as the failure to distinguish between understanding the semantics and syntax of a diagram and the lack of feedback given to subjects about their understanding (Burton-Jones and Meso, 2002). We adopted several strategies to avoid these
difficulties. First, we included items that asked subjects to what extent they understood the diagrams as a whole as well as to what extent they understood the particular parts of the diagram affected by the treatment. For example, an item measuring perceived ease of understanding is "to what extent is the information represented by properties (e.g. hasPreference) that are connected to classes in the diagrams easy to understand?" Second, we ensured that our dependent measures focused on the extent to which subjects understood the semantics of the diagrams rather than the layout or syntax. We suggest that understanding derived from information represented in the diagram and not from the layout of the diagram. Third, we provided subjects with feedback on their performance in the knowledge identification tasks prior to them answering the questions on perceived variables. People often use feedback to increase their perceived understanding (Cahn and Frey, 1989). The feedback (a set of correct answers of the knowledge identification tasks) was prepared with the help of two domain experts. By comparing the correct answers in this feedback with the answers they gave, subjects should be able to more correctly assess the degree to which they understood the diagrams.

4.5.2.3 Control variables

Although random assignment controls for most internal validity threats (Cook and Campbell, 1979), we obtained data on several control variables to provide additional confidence that differences between groups stemmed from the experimental treatment rather than from some confounds. These variables were: domain knowledge, modeling knowledge, time to perform the task, order of the domains, and perceived ease of syntax understanding.

Domain knowledge is used as a control variable because domain experts might perform knowledge identification better by simply referring to their prior knowledge (Pretz et al., 2003). Items regarding domain familiarity and domain experience were used to measure this variable. Modeling knowledge was measured because subjects with high modeling knowledge may find the tasks easier to perform than subjects with low modeling knowledge. Items regarding modeling concepts familiarity and modeling experience were used to measure this variable. Time was measured because subjects performing knowledge identification with a poor representation

26 Given the tasks (see below) we obtained the help of a travel agent and an experienced auction website user.
may be able to overcome the problems in the representation by spending longer on the exercise (Jarvenpaa and Machesky, 1989). Finally, to test whether the layout of the diagrams made one diagram easier to understand than the other we used a scale of perceived ease of understanding (syntax) proposed by Gemino (1998) as a control variable.

To create conservative, precise empirical tests, several studies in conceptual modeling recommend that researchers who test the extent to which individuals understand different diagrams should ensure that the diagrams are “informationally equivalent” (Gemino and Wand, 2005). If one diagram contains more information than another diagram, the results of empirical tests may be obvious and uninteresting (Parsons and Cole, 2005). We followed this advice when creating the diagrams for our experiment by ensuring that the informed and uninformed ontologies contained the same information; they only differed in the degree to which information about interactions was explicit or implicit. We claim that the two types of diagram are informationally equivalent if the informed one can be created solely based on the uninformed one and the guidelines. To make sure that every concept (such as classes and properties) mentioned in one diagram can be identified from the other, we developed mapping tables (Table 20-24, Appendix G shows all such mappings).

To provide a manipulation check for the informational equivalence in the experiment, we included a measure of diagram comprehension (Gemino, 1998). The comprehension questions required subjects to answer simple yes/no questions about the diagrams and were drafted carefully to cover the entire diagrams.

Appendix H details the questions related to domains used in the study to measure the dependent variables and control variables.

4.5.3 Experimental setting

4.5.3.1 Experimental Procedure

A 1*2 between-group design was used. Each group received informed or uninformed OWL diagrams, with subjects randomly assigned to groups. Subjects first received training on OWL

\textsuperscript{27} i.e. without using any additional knowledge.
concepts and practiced answering knowledge identification questions. Subjects received feedback on the performance of the practice task. They were then given an ontology diagram (informed or uninformed) and answered comprehension questions followed by knowledge identification questions. The sequence was repeated for a second domain. The order of domains was randomized. At the end, subjects compared their answers of the knowledge identification tasks with a set of correct answers (feedback) and then completed a post-test questionnaire measuring the perceived variables (understanding and ease of understanding).

As the study involved human subjects, an ethical approval was obtained from the University (Appendix I)

4.5.3.2 Selection of domains and ontologies

Pretz et al. (2003) claim that when faced with a problem-solving question, different individuals create different problem representations based on their prior domain knowledge because such knowledge mediates an individual’s ability to represent the problem in the most efficient fashion. Individuals who have accumulated considerable knowledge in a domain, represent information about problems differently than individuals who do not have such extensive knowledge bases (Chi et al., 1988). Subjects who have expertise in the domain (high domain knowledge) may not refer much to the problem presentation described by the ontology diagram as their mental representations of the problem would be guided by their prior domain knowledge. On the other hand, if the subjects are completely unfamiliar with the domain, they will have difficulty in understanding the information in the domain presentation. In either case, the effect of ontological diagrams in describing the domain cannot be established. Therefore, the domains under study were selected carefully so that subjects would not be too familiar or too unfamiliar with these domains.

To increase external validity and reduce researcher bias, we selected ontologies that had already been developed and used in practice. Two ontologies were used: a travel ontology and an auction ontology. The ontologies were published (in formal OWL notation) at www.schemaweb.info, which is a non-profit portal intended to share ontologies. These ontologies had been developed and used for research on agent based services (Zou, 2004), and thus had been evaluated for other purposes. Because these ontologies had been created without
our proposed guidelines, they served to create the "uninformed" diagrams in the experiment. We then applied our proposed guidelines for modeling interactions to these ontologies to create the "informed" versions of them. When we first applied our guidelines to these diagrams, we found that this resulted in the informed diagrams actually containing more information than the uninformed diagrams. Consistent with the recommendations discussed above, we therefore added information to the "uninformed" diagrams to make them informationally equivalent. The additions were in the form of new classes or added attributes for existing classes in the original ontologies. Note that this makes our test "conservative" i.e., our results will understate the benefit that can be obtained by following our proposed modeling rules. After creating the ontologies, the knowledge identification questions were developed.

The descriptions of these ontologies are provided in Appendix E and the visual representations of these ontologies are provided in Appendix F (Figure 39-44).

4.5.3.3 Subjects

As the area of practice of knowledge identification task is still emerging, it would have been nearly impossible to find a representative set of KMS users who use visual representation of ontologies. Thus, as a substitute, our subjects were undergraduate business students. These students had previously taken one or more database-oriented courses and thus were relatively familiar with concepts such as classes and properties and with the use of visual representations.

4.5.3.4 The pilot study

To refine the instruments measuring dependent and control variables, a pilot study was conducted with 22 subjects. As Table 9 shows, the results support hypothesis 1. For hypotheses 3, the results are in the hypothesized direction, but it is not statistically significant.

<table>
<thead>
<tr>
<th>Dependent Var.</th>
<th>df</th>
<th>Mean (Standard Deviation) between group</th>
<th>F</th>
<th>Sig. (1-tail)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNOWID</td>
<td>20, 1</td>
<td>24.55 (11.33) (Informed) &gt; 16.64 (3.44) (Uninformed)</td>
<td>4.90</td>
<td>.039*</td>
</tr>
<tr>
<td>PEOU</td>
<td>20, 1</td>
<td>33.00 (5.02) (Informed) &gt; 30.91 (3.39) (Uninformed)</td>
<td>1.31</td>
<td>.266</td>
</tr>
<tr>
<td>PU</td>
<td>20, 1</td>
<td>20.55 (2.50) (Uninformed) &gt; 19.09 (3.04) (Informed)</td>
<td>1.49</td>
<td>.236</td>
</tr>
</tbody>
</table>

Table 9: One way ANOVA for differences Between Groups
We ran an ANCOVA to test for the possible effects on knowledge search performance of the control variables: modeling knowledge (MODKNO), domain knowledge (DOMKNO), perceived ease of syntax understanding (PEOU_SYN) and time taken for knowledge search tasks (KI_TIME). As Table 10 shows, none of these variables affected the performance of the knowledge search task.

<table>
<thead>
<tr>
<th>Variables</th>
<th>df</th>
<th>F</th>
<th>Sig. (1-tail)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODKNO</td>
<td>1</td>
<td>.687</td>
<td>.419</td>
</tr>
<tr>
<td>DOMKNO</td>
<td>1</td>
<td>.238</td>
<td>.632</td>
</tr>
<tr>
<td>PEOU_SYN</td>
<td>1</td>
<td>2.73</td>
<td>.118</td>
</tr>
<tr>
<td>KI_TIME</td>
<td>1</td>
<td>.119</td>
<td>.735</td>
</tr>
<tr>
<td>TREAT</td>
<td>1</td>
<td>6.272</td>
<td>.023*</td>
</tr>
</tbody>
</table>

Table 10: ANCOVA analysis for knowledge identification tasks

Overall, the results suggest that our research question can be answered in the affirmative, but only in terms of subjects’ actual performance in knowledge search tasks. We found no significant difference on the perceived variables PEOU and PU. We recognize that this may be due to a lack of power and that the effect may become significant as the sample size is increased.

Based on the pilot study, several improvements were made to the final study. First, it was observed that item reliability for perceived ease of understanding was low (α = 0.60). The original instrument had seven questions – four related to the whole diagram and three to specific treatment items. Factor analysis indicated that items related to the treatment loaded separately with low factor loadings. Item reliability analysis of perceived ease of use -Syntactic was also low (α= 0.46). These results indicated that items related to parts of the diagram affected by the
treatment did not reflect perceived ease of understanding. This finding might be an indication that subjects did not understand items related to only parts of diagram (those affected by the treatment). Thus it was decided to eliminate all perceived variables related to parts of the diagrams and use only those related to the whole diagram.

Second, we found that subjects who only had introductory exposure to database concepts exhibited low performance in comprehension questions (score of 12 out of 18, \( n = 14 \)). This indicated that they might not have understood the ontological diagrams sufficiently well to perform the task. Thus, we decided to only include in the final study subjects who had taken at least two database-related courses. Third, we found that the list of correct answers that was provided to subjects as feedback was incomplete. Additional correct responses were added to the feedback to remedy this problem.

4.6 Results and Discussion

Based on these improvements, a final study was conducted with 56 subjects. The descriptive statistics (Table 11) show that scores of knowledge identification tasks were high (an average of 3.2 correct responses out of 5 per task). This could be attributed to the training provided to the subjects at the beginning of the study. The descriptive statistics also show that subjects had higher modeling knowledge (mean 3.81) than domain knowledge (3.40). This was expected as the subjects were not too familiar with the domains and they had taken two courses that contained database content. The descriptive statistics also showed that there were no significant problems of normality in the data.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Mean</th>
<th>St. Dev.</th>
<th>Range</th>
<th>Skew</th>
<th>Kurt</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNOWID</td>
<td>19.2</td>
<td>7.3</td>
<td>35</td>
<td>.26</td>
<td>.04</td>
</tr>
<tr>
<td>PU</td>
<td>4.48</td>
<td>0.85</td>
<td>4</td>
<td>-0.07</td>
<td>-.01</td>
</tr>
<tr>
<td>PEOU</td>
<td>4.11</td>
<td>0.74</td>
<td>3</td>
<td>0.58</td>
<td>-.17</td>
</tr>
<tr>
<td>MODKNO</td>
<td>3.81</td>
<td>1.31</td>
<td>6</td>
<td>0.55</td>
<td>0.02</td>
</tr>
<tr>
<td>DOMKNO</td>
<td>3.40</td>
<td>1.30</td>
<td>5</td>
<td>0.12</td>
<td>-0.93</td>
</tr>
</tbody>
</table>

KNOWID: Knowledge identification: number of correct answers (open), PU: Perceived Understanding (out of 7), PEOU: Perceived Ease of Understanding (out of 7), MODKNO: Modeling Knowledge (out of 7) and DOMKNO: Domain Knowledge (out of 7), \( N=56 \), 2 groups * 28 per cell

Table 11: Descriptive Statistics for the variables

To measure the instrument validity and reliability, several statistics were used (Table 12).
107

<table>
<thead>
<tr>
<th>Variables</th>
<th>Validity</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNOWID</td>
<td>Tested in MANOVA (see Table 15)</td>
<td>Inter-rater reliability: ( r = 0.79-0.89 )</td>
</tr>
<tr>
<td>PU</td>
<td>Factor loadings: 0.88, 0.92, 0.90</td>
<td>Cronbach’s alpha 0.88</td>
</tr>
<tr>
<td>PEOU</td>
<td>Factor loadings: 0.86, 0.75, 0.84, 0.51</td>
<td>Cronbach’s alpha: 0.54</td>
</tr>
</tbody>
</table>

Variables: KNOWID: Knowledge identification, PU: Perceived Understanding, and PEOU: Perceived Ease of Understanding

Table 12: Instrument validity and reliability

The perceived understanding scale appeared to be valid as individual items of this scale loaded highly on the construct and the alpha value was more than 0.7 (Nunnally, 1967). The alpha value for perceived ease of understanding was relatively low (0.54). On closer inspection, we noticed that one item of this scale ("the diagram required lot of mental effort") had low factor loading (0.51). On removing this item from the scale, the alpha value was 0.85 and factor loadings of the rest three items were >0.80 (.82, .83, .86). Thus, after removing this item, the two scales were considered to have acceptable reliability.

We also ran a factor analysis of all perceived items together. The result indicates that the items loaded distinctively in two separate perceived variables (for PEOU < 0.5 and for PU > 0.8), indicating that the two groups of items measured different constructs.

To obtain high internal validity, it is necessary that the two types of representation be informationally equivalent (Parsons and Cole, 2005). As a manipulation check, we used subjects’ comprehension scores to test whether the ontological representations (informed and uninformed) were informationally equivalent. The mean difference for comprehension scores was small (15.43 for uninformed ontologies and 15.36 for informed ontologies). The scores on comprehension tasks using informed and uninformed ontology diagrams were found not to be significantly different (\( F = 0.037, p = 0.847 \)). Also, we found no significant difference on the time taken by subjects in answering the comprehension questions (\( F = 0.365, p = 0.549 \)). Thus, information equivalence could be assumed.

Following Burton-Jones and Meso (2006), we used inter-rater reliability and MANOVA to test the reliability and validity of knowledge identification measurements respectively. To measure inter-rater reliability of the coded data we adopted strategies prescribed by Khatri et al.
Two MIS PhD students (who were not involved otherwise in the research and who had good knowledge of conceptual modeling) were hired to code the knowledge identification data. Each coder independently coded all the responses. A detailed coder's manual was prepared to facilitate the coders to code the tasks, consisting of detailed coding procedures and examples of sample coding (Appendix J). To ensure that the coders had a common basis to score each response, the coders had to first evaluate for each response: (a) whether a procedure can be developed from the response and (b) whether the response relates to the particular knowledge identification task asked. Following this, the coders had to allocate 0 or 1 for each response. Finally, the coders responded in a 1 to 3 scale (1-Low and 3-High) how confident they were in evaluating the responses. We also controlled the "coder drift" phenomenon - the tendency to change the way in which coding definitions are applied over time (Khatri et al. 2006). To control this drift, each coder was given a set of sample responses to be coded individually before coding the responses in batches. The primary researcher discussed the results of this sample coding with the coders and resolved any differences in coding.

Based on this stringent coding procedure, the inter-rater reliability was relatively high (range 0.79 and 0.89). The confidence ratings of coders were also high (Table 13).

<table>
<thead>
<tr>
<th>Coders</th>
<th>1 (Low)</th>
<th>2 (Medium)</th>
<th>3 (High)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coder A</td>
<td>4%</td>
<td>9%</td>
<td>87%</td>
</tr>
<tr>
<td>Coder B</td>
<td>6%</td>
<td>26%</td>
<td>68%</td>
</tr>
<tr>
<td>Average</td>
<td>5%</td>
<td>17%</td>
<td>77%</td>
</tr>
</tbody>
</table>

Table 13: Confidence Rating of Coders

We also ran a MANOVA to study the effect of each item on the knowledge identification task. The results support the convergence validity of the knowledge-search tasks as there were significant differences on every question in the two tasks (see Table 14).
Table 14: MANOVA analysis for individual items of knowledge identification task

<table>
<thead>
<tr>
<th>Knowledge identification</th>
<th>Uninformed mean (SD)</th>
<th>Informed mean (SD)</th>
<th>F</th>
<th>Sig.</th>
<th>Part. Eta Sq.#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auction Item 1</td>
<td>5.96 (3.1)</td>
<td>9.26 (3.25)</td>
<td>15.25</td>
<td>.000</td>
<td>.22</td>
</tr>
<tr>
<td>Auction Item 2</td>
<td>3.75 (1.75)</td>
<td>5.61 (2.29)</td>
<td>11.54</td>
<td>.001</td>
<td>.17</td>
</tr>
<tr>
<td>Auction Item 3</td>
<td>4.68 (2.29)</td>
<td>6.39 (2.28)</td>
<td>7.85</td>
<td>.007</td>
<td>.12</td>
</tr>
<tr>
<td>Travel Item 1</td>
<td>5.68 (3.11)</td>
<td>9.32 (3.67)</td>
<td>16.02</td>
<td>.000</td>
<td>.22</td>
</tr>
<tr>
<td>Travel Item 2</td>
<td>4.43 (2.34)</td>
<td>6.21 (2.84)</td>
<td>6.55</td>
<td>.013</td>
<td>.10</td>
</tr>
<tr>
<td>Travel Item 3</td>
<td>5.43 (2.61)</td>
<td>8.46 (3.65)</td>
<td>12.76</td>
<td>.001</td>
<td>.19</td>
</tr>
</tbody>
</table>

*Eta square reflects the proportion of variance in the dependent variable explained by the treatment.  
NOTE: The overall MANOVA for the six items was significant at F (Pillai’s trace) = 10.84, p = .00.

Table 14: MANOVA analysis for individual items of knowledge identification task

To test the hypotheses we conducted one-way ANOVA. The results support all the three hypotheses (Table 15).

<table>
<thead>
<tr>
<th>Dependent Var.</th>
<th>df</th>
<th>Mean (Standard Deviation) between group</th>
<th>F</th>
<th>Sig. (1-tail)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNOWID</td>
<td>54</td>
<td>23.00 (6.78) (Informed) &gt; 15.39 (5.72) (Uninformed)</td>
<td>20.60</td>
<td>.000*</td>
</tr>
<tr>
<td>PU</td>
<td>54</td>
<td>4.83 (0.78) (Informed) &gt; 4.13 (0.77) (Uninformed)</td>
<td>11.43</td>
<td>.111</td>
</tr>
<tr>
<td>PEOU</td>
<td>54</td>
<td>4.52 (0.58) (Informed) &gt; 3.70 (0.65) (Uninformed)</td>
<td>24.54</td>
<td>.000*</td>
</tr>
</tbody>
</table>

KNOWID: Knowledge identification, PU: Perceived Understanding, PEOU: Perceived Ease of Understanding,
*Significant at .05 level

Table 15: ANOVA analysis for hypotheses testing

To test the possible effects of the control variables, we performed an ANCOVA. The results (Table 16) indicate that control variables had no significant effect; only the experimental treatment (TREAT) had a significant effect.

<table>
<thead>
<tr>
<th>Variables</th>
<th>df</th>
<th>F</th>
<th>Sig. (1-tail)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODKNO</td>
<td>1</td>
<td>1.156</td>
<td>.288</td>
</tr>
<tr>
<td>DOMKNO</td>
<td>1</td>
<td>.024</td>
<td>.879</td>
</tr>
<tr>
<td>PEOU_SYN</td>
<td>1</td>
<td>2.639</td>
<td>.111</td>
</tr>
<tr>
<td>KI_TIME</td>
<td>1</td>
<td>.365</td>
<td>.549</td>
</tr>
<tr>
<td>DOM_ORD</td>
<td>1</td>
<td>.228</td>
<td>.635</td>
</tr>
<tr>
<td>TREAT</td>
<td>1</td>
<td>15.608</td>
<td>.000*</td>
</tr>
</tbody>
</table>

Variables: MODKNO: Modeling knowledge, DOMKNO: Domain Knowledge, PEOU_SYN: Perceived Ease of Use- Syntactic, KI_TIME: Knowledge Identification time, DOM_ORD: Domain order and TREAT: Treatment  
* Significant at .05 level

Table 16: ANCOVA analysis for knowledge identification tasks
Overall, the results suggest that the research question can be answered affirmatively, i.e., a visual representation of a domain ontology that complies with ontological guidelines will provide better support for knowledge identification than a visual representation of a domain ontology that does not comply with ontological guidelines.

4.6.1 The effect of language in the results of the study

In this study languages are used at several levels. First, through the use of a high level ontology (Bunge’s ontology), second, use of a specific formalization represented in an ontology language (OWL), and third, through tasks that are for specific contexts. This raises a question whether the use of languages at these different levels restrict the scope and findings of the research? This section addresses this question.

To understand this concern, the approach based on Wittgenstein’s philosophy of language is discussed. According to Rorty (in the book Linguistic Turns, 1967), linguistic philosophy views the philosophical problems as problems which may be solved either by reforming language or by understanding more about the language that we presently use. Linguistic philosophers talk about the world by means of talking about a suitable language—this is the linguistic turn (Rorty, 1967). The basic proposition of linguistic philosophy is through the description in language the world is revealed to us (Hirschheim et al., 1995). Thus the world (reality) is never given to us in and of itself, but only through interpretation in some language. This can be further interpreted as when two observers are confronted with identical observations, they will not draw the same conclusions unless their language can somehow be calibrated.

To address this concern the following steps and assumptions are made.

1. A fundamental assumption made in this study is that the real world exists irrespective of the human cognizance and interpretation. This is a realist-based position. Other approaches that deny this position can be used to conduct such studies.

2. An assumption made in this study is that the subjects have a common understanding of the notions of – class, properties and interactions. This understanding has developed by the use of language in their day-to-day life. As these concepts are very generic in nature and the study is conducted in one language (English) that all the subjects are familiar to,
therefore it is assumed that the interpretation of these notions by all subjects is more or less the same.

3. The empirical study uses two domains (auction and travel) that subjects are somewhat familiar with. If the domains were too unfamiliar then the concepts of the domain (in the form of properties and classes) could have been interpreted differently by different subjects. The role of language for such interpretations might have been a factor in that case.

4. Several terminologies (such as itinerary, auction) were used in the ontologies. Inconsistent interpretation of these terms might have confounded the results of the study. Thus to check whether subjects understand these terms consistently, four subjects having different backgrounds were asked to define these terms without referring to any documents (such as dictionary). These definitions were then compared with definitions obtained from the oxford dictionary (Table 25 in Appendix K). The table indicates that subjects interpreted all the terms consistently.

5. Finally, to test the hypotheses two domains (instead of one) were used. This helped to generalize that the results are applied on multiple contexts.

4.7 Conclusion

The findings of this study have theoretical, methodological, and practical implications. We now discuss those.

From a theoretical point of view, the work has several implications. First, it defines and highlights the importance of knowledge identification in the context of KMS use. Second, it suggests a distinction between knowledge identification and problem solving and formally defines the former. Third, it extends the information search literature by identifying a specific role of ontology in the context of knowledge search. Fourth, the paper suggests an approach to improving the expressiveness of formal ontologies.

Methodologically, the paper presents an operationalization of knowledge identification tasks. Specifically, the questions about required knowledge are framed in the context of developing procedures to attain certain goals. This approach facilitated the development of an instrument to measure knowledge identification performance. As well, the research employed
several additional steps to those used in similar studies (those comparing conceptual modeling techniques). First, the study introduced a technique to test for informational equivalencies of diagrams. Second, the perceived understanding and perceived ease of understanding measurements were designed to focus on semantics of the diagrams rather than the layout or syntax. Third, feedback on subject performance was used prior to answering questions related to the perceived understanding and perceived ease of understanding.

The outcomes of the research can also have practical implications. Industry reports predict that by 2007, 60% of information access implementations will combine search, ontology, and information visualization technologies (Gartner, 2005b). Rao and Osei-Bryson (2006) suggest that a formal ontology used to organize the knowledge in the knowledge base of a KMS can also be accessed directly by the end user. In this vein, we suggest that users should be able to search and browse ontologies to perform knowledge identification. We found significant differences between subjects using alternative visual ontologies on all of our dependent measures. This indicates that users can both derive benefit from ontology diagrams designed to convey domain knowledge and that they perceive this benefit. This finding suggests that KMS designers need to pay attention to visual representations of the ontologies underlying the knowledge managed by the KMS. Moreover, the results also suggest that the proposed guidelines can be helpful in the practice of developing visual formal ontologies.

Related to the finding, we note that we might have used another visual ontology representation (other than OWL) or another philosophical ontology as guidance. However, we claim that what matters is the demonstration that the type of visual ontological representations can make a difference in performing knowledge identification tasks and that the use of a philosophical ontology as a guidance can be advantageous.

We discuss now some limitations of our study. First, the research is more relevant to users who have limited domain understanding and at least some modeling knowledge. It is difficult to define how much domain knowledge or modeling knowledge a user will have in practice. Second, our study was limited to comparing ontologies developed with ontological guidelines with ontologies developed without guidelines. It would have been more effective if we were able to compare our ontological guidelines with some existing ones. But as no such
guidelines currently exist for creating formal ontologies, we had to depend on ontologies that were developed based on current practices. Third, our focus in this study was more on gaining internal validity than on external validity as we conducted a controlled laboratory experiment and limited our study to measure the effect of one treatment (i.e. the interaction modeling rules). We note that by drawing experimental materials (diagrams) from practice, we were able to maintain some external validity. Fourth, due to practical constraints, we used student subjects in this study. However, this is common practice employed in much of the preceding conceptual modeling research.

Several future research opportunities emerge from this study. First, research can be directed to test the effect of more complete guidelines on knowledge identification. In such research, additional domain concepts can be modeled explicitly in OWL resulting in more guidelines. An example of such a concept is - composite things. Composites are things made of other things (Shanks, Tansley, and Weber, 2004) and abundant phenomena exist that can be modeled by them (e.g. product compositions, departmental structures and project teams). Second, our study did not investigate why users benefited from using the informed ontologies. This question may be studied by employing process tracing methods, in particular, those examining difficulties or breakdowns experienced by subjects using diagrams (Burton-Jones and Meso, 2006). Third, external validity might be improved by providing subjects with a more realistic environment in the form of a “mock up” KMS interface. In such an environment, subjects will be able to examine and browse the diagrams. Fourth, currently, very limited methodological support is available for developing ontologies (Gomez-Perez et al., 2004). Therefore, it would be of interest to test whether the guidelines will be of actual use to ontology developers. Fifth, as we have indicated above, we used a particular philosophical ontology to derive construction guidelines for formal ontologies. It would be of interest to conduct comparative testing of formal ontologies developed according to different philosophical ontologies.
References


5.1 Review

Although substantial interest exists in knowledge management systems (KMS), a theoretically-based view of knowledge in the KMS context is not yet available. The literature still presents contrasting and at times even contradictory views of knowledge. Chapter 2 presents the development of a conceptual framework for knowledge in the KMS context. The key concepts of the framework are derived by combining an action-based perspective with an artificial intelligence (AI) view of knowledge. The relationships among these concepts are identified by analyzing current literature about knowledge. The concepts and their relationships form the basis for three conceptual models – the first describes knowledge from an agent’s perspective, the second defines knowledge as used by an agent, and the third represents the role of KMS in providing access to knowledge. The models are presented graphically and related to current knowledge management (KM) literature. Implications of the models for the development of KMS are discussed. Finally, the paper suggests ways to test the proposed view of knowledge.

In chapter 3 it is suggested that the use of philosophically-based ontological principles can help generate guidelines for developing conceptual models using OWL. Conceptual models are used to support the understanding of and communicating about application domains in the development of computerized information systems. Such models are created using modeling grammars (which often can be represented graphically). Ontology description languages such as OWL (Web Ontology Language) have been proposed for creating such formal representations.

To be effective, a grammar should be able to convey the meaning of the domain concepts and their relationships. In chapter 3 it is proposed that language grammars developed for defining formalized ontologies can be used, with appropriate modeling guidelines, to create meaningful conceptual models. The main requirements from ontology description languages are that they have well-formalized syntax and that they will be computer-readable. However, not much attention has been paid to how they can be used to convey domain semantics. The chapter
presents specific guidelines for creating conceptual models in OWL and, via examples and a case study, demonstrates how the application of the guidelines can provide clearer representation of domain phenomena.

To use Knowledge Management Systems (KMS) effectively, knowledge seekers need to be able to identify the knowledge required to perform their task. Knowledge identification can be facilitated by providing users with information about the relevant concepts underlying the problem domain. A formal set of statements defining concepts and their relationships is termed a formal ontology. Formal ontologies are designed for software applications rather than to support domain understanding. Therefore, they might not be able to convey domain information properly. It is suggested that the usefulness of formal ontologies can be enhanced by constructing them according to guidelines reflecting a philosophically based ontology. Formal ontologies developed with such guidelines are termed as informed ontologies. Chapter 4 describes a laboratory study that compared the use of visual forms of informed and uninformed ontologies. The purpose was to test whether the use of visual representations of informed ontologies would lead to better knowledge identification than the use of uninformed ontologies. The results indicated that the use of informed ontologies was advantageous in terms of helping subjects to identify knowledge needed to perform a task. Furthermore, subjects perceived informed ontologies to be more understandable than the uninformed ones. The main conclusion is that visual information can help in knowledge identification and that attention should be paid to how the representation conveys domain information.

To develop the thesis chapters, several positions are taken. These are: (1) an ontological point of view of knowledge, (2) the use of a realist ontology to define fundamental concepts, (3) a pragmatic approach to the meaning and role of knowledge and (4) a focus on knowledge as possessed by individuals. These positions are described below:

An ontological point of view of knowledge: ontology is used to describe knowledge concepts. Such approach is based on the belief that there exist some fundamental phenomena in reality that help to describe knowledge.

A pragmatic approach to the meaning and role of knowledge: the two tenets of this approach are: (1) the role of knowledge is derived on how it is used in practice and (2) knowledge is rooted in
actions. In the pragmatic approach, knowledge is considered not just as a theoretical concept but anchored to how it is used specially in the context of KMS.

A focus on knowledge possessed and used by individuals: the knowledge management literature distinguishes knowledge as possessed and used by an individual from knowledge acquired and shared by a group. The processes by which a group acquires, shares, transfers, and uses knowledge is often referred to as knowledge management processes. Chapter 2 focuses on the individual view of knowledge as this view should precede the process view of knowledge. This is because it is important to clarify what is knowledge before discussing how it can be acquired, shared or transferred. However, it is to be noted that it might be possible to develop conceptual models of knowledge management processes using individualistic knowledge models presented in chapter 2.

The use of a realist ontology to define fundamental concepts: realists believe that reality exists independent of our awareness. Dobson (2002) discusses the realist position as follows “Our perceptions of reality change continually but the underlying structures and mechanisms constituting that reality are "relatively enduring.” The aim of realist research [in the context of IS] is to develop a better understanding of these enduring structures and mechanisms. Ontological factors, therefore, must be the primary factor in defining research approaches.” Accordingly, the reason for choosing a realist ontology is that it helps us to understand the concepts of the real world. This approach is specifically taken in the chapter 3 where concepts of a realist ontology are used to develop guidelines for using OWL as a conceptual modeling language.

5.2 Contributions

This thesis extends the current research in several ways. First, it provides a theoretical framework of knowledge in the context of KMS. By providing such framework, the role of KMS in managing knowledge is clarified. Such clarification is helpful in suggesting design principles of KMS. Second, languages to develop formal ontologies are suggested to be used as conceptual modeling languages. Such suggestion helps to represent domain facts accurately. Third, visual ontologies are suggested to assist performing knowledge identification tasks and finally, the
usefulness of formal ontologies in performing knowledge identification tasks is evaluated empirically.

Overall there are two main contributions of this research. First, an important use of IS ontologies in the context of KMS is suggested. Currently, IS ontologies are proposed and used in KMS in two ways. First, to extract knowledge from unstructured and semi-structured information and then store it in KMS, and second, to facilitate knowledge access and sharing in KMS (Davies et al., 2003). In this research, it is suggested and empirically corroborated that IS ontologies can be used to help individuals perform knowledge identification. Second, the research contributes in providing a theoretical framework of knowledge in the context of KMS. Such framework is important in order to clarify the concept of knowledge in the KMS context.

The contributions of this thesis can be categorized into theoretical, practical, and methodological. These are as follows:

**Theoretical**
1. Clarified the term “knowledge” in the context of KMS
2. Defined the terms “knowledge identification”, “knowledge tagging”, and “knowledge query” in the context of KMS
3. Suggested a distinction between knowledge identification and problem solving
4. Extended the information search literature by identifying a specific role of ontology in the context of knowledge search
5. Differentiated the roles of KMS and IS
6. Suggested how to use OWL as a conceptual modeling language

**Methodological**
7. Suggested how knowledge can be defined in the context of KMS using concepts from action based philosophy and AI literature.
8. Suggested the use of visual ontologies to support knowledge identification
9. Suggested how OWL can be used as a conceptual modeling language by referring to concepts of philosophical ontologies
10. Suggested how to operationalize “knowledge identification” for conducting an empirical study. In particular, developed an instrument to measure knowledge identification performance.

11. Introduced improvements for conducting empirical studies on comparing conceptual modeling techniques, specifically:
   a. Introduced a technique to test informational equivalences of diagrams
   b. Introduced feedback on subject performance prior to answering questions related to perceived and perceived ease of understanding.

12. Suggested how to test “knowledge description” and “knowledge tagging”.

13. Suggested guidelines on how to identify, tag, and query knowledge from KMS. Some examples on how to use these guidelines are provided in chapter 3.

14. Suggested that the guidelines prescribed to use OWL as conceptual modeling language can be implemented in the form of a case tool. Such a tool can guide modelers to model domain concepts.

5.3 Future Work

Each chapter described in this thesis opens the possibility of conducting new research. Chapter 2 presented a framework for knowledge in the context of KMS. This chapter can be extended in two ways. First, conceptual models of knowledge management processes (knowledge acquired, shared, and transferred by a group) can be developed using the conceptual models of knowledge. Development of such models might provide insight on the role of KMS in the KM activities. Second, several tests (described in section 2.6) can be suggested to test the proposition that for an individual agent, knowledge can be viewed in terms of current state, goal state, and possible actions.

The research described in chapter 3 can be extended in several ways. First, other philosophical ontologies (other than Bunge's ontology) can be used as a philosophical base. Second, other domain phenomena such as composites, laws, states and events for which there is no direct representation in OWL, exist Future research can explore mechanisms to model such additional constructs in OWL. Third, based on the analysis done in the chapter, it is possible to
extend and modify OWL. This, in turn, can help constructing/modifying ontology development tools (such as Protégé) in OWL to facilitate the development of domain concepts such as interactions and composites.

Several empirical studies can be suggested that can extend the research described in chapter 4. First, an empirical study (in the form of process tracing) can be done to investigate why users benefit from using informed ontologies to perform knowledge identification. Second, related to the empirical study (in chapter 4), a study of the effects of more complete guidelines on knowledge identification can be performed. The empirical study described in chapter 4 uses the guidelines on how to represent interactions using OWL. Guidelines to develop other domain facts (such as composites) can be used to create more complete guidelines. Subsequently, the effect of these guidelines can be tested using a similar study as described in chapter 4. Third, empirical study can be designed to test whether the guidelines can be of actual use to ontology developers. In this proposed study, the usability and effectiveness of ontologies created with the modeling guidelines can be evaluated. Finally, comparative tests can be performed on formal ontologies developed based on different philosophical ontologies.
References


Appendix A: Organizational Knowledge Definitions

<table>
<thead>
<tr>
<th>Organizational knowledge definitions</th>
<th>A</th>
<th>C</th>
<th>S</th>
<th>G</th>
<th>I</th>
<th>Concepts not mapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge is defined as a justified belief that increases an entity’s capacity for effective action (Huber, 1991).</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Belief</td>
</tr>
<tr>
<td>Knowledge is a personal capacity that should be seen as the product of the information, experience, skills and attitude which someone has at a certain point in time (Weggerman, 1997).</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Experience, skills, attitude</td>
</tr>
<tr>
<td>Knowledge is essentially related to action and information becomes knowledge when it is interpreted by individuals and given a context and anchored in the beliefs and commitments of individuals (Nonaka and Takeuchi, 1995).</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Context, belief</td>
</tr>
<tr>
<td>Knowledge comprises all cognitive expectancies, observations that have been meaningfully organized, accumulated and embedded in a context through experience, communication or inference- that an individual or organizational actor uses to interpret situations and to generate activities, behavior and solutions no matter whether these expectations are rational or used intentionally (Maier, 2002).</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Expectancies, observations, experience, context</td>
</tr>
<tr>
<td>Knowledge is a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and Information. It originates and is applied in the minds of knowers. (Davenport and Prusak, 1998).</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Experience, contextual information</td>
</tr>
<tr>
<td>Knowledge is information processed in the mind of individuals; it is personalized information related to facts, procedures, concepts, interpretations, ideas, observations, and judgments. (Alavi and Leidner, 2001).</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Facts, procedures, concepts, interpretations, ideas, observations, and judgments</td>
</tr>
<tr>
<td>Knowledge is information whose validity has been established through test of proof (Liebeskand, 1996)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Knowledge can be seen as some basic human needs to structure, categorize and interpret the world around them in certain ways. It is the need to reduce uncertainty. (Beijerse, 1999).</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Knowledge can be defined as the capacity and the ability to act (Sveiby, 1997).</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Capacity, ability</td>
</tr>
<tr>
<td>Knowledge is internalized know-how, the ability to tacitly know—in any particular context—what needs doing and how it should be done (May and Taylor, 2003).</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Know-how, ability, context</td>
</tr>
<tr>
<td>Knowledge is a meaningful set of information that constitutes a justified true belief and/or embodying a technical skill through practice (Nonaka et al., 1996).</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Belief, skill</td>
</tr>
</tbody>
</table>
Knowledge is about imbuing data and information with decision and action relevant meaning (Fahey and Prusak, 1998).
Knowledge is information that is relevant, actionable, and based at least partially on experience (Leonard and Sensiper, 1998).
Knowledge is an ongoing social accomplishments constituted and reconstituted in everyday practice (Orlikowski, 2002).
Knowledge is an organized combination of data, assimilated with a set of rule, procedures, and operations learnt through experience and practice (Bhatt, 2001).
Knowledge is contextual and includes an actionable summary and interpretation of experience (Amaravadi and Lee, 2005).
Knowledge consists of the assimilation of related information addressed in the context of reference (Marsh, 1997).
Knowledge can be defined as the situation where insight is achieved in a context by pointing out information from data as the difference that makes a difference (Jensen, 2005).
Knowledge is information whose validity has been established through test of proof (Liebeskind, 1996).
Knowledge is a clear understanding of information and their associated patterns (Bierly et al., 2000)
Knowledge is a collection of information and rules with which a certain function can be fulfilled (Hertog and Huizenga, 2000).
Knowledge is meaningfully organized accumulation of information through experience, communication or inference (Bobrow and Collins, 1975).
Knowledge is that which is known and so it is an essentially human form of information (Wilson, 1996).
Knowledge consists of truths and beliefs, perspectives and concepts, judgments and expectations, methodologies and know-how (Wiig, 1993).
Knowledge is a collective activity embodied in situated practice (Lave and Wenger, 1991).
Information used to take decisions forms knowledge upon which people base actions to achieve results (Davenport et al., 2001)

<table>
<thead>
<tr>
<th>Knowledge</th>
<th>A</th>
<th>C</th>
<th>S</th>
<th>G</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge is about imbuing data and information with decision and action relevant meaning (Fahey and Prusak, 1998).</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge is information that is relevant, actionable, and based at least partially on experience (Leonard and Sensiper, 1998).</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge is an ongoing social accomplishments constituted and reconstituted in everyday practice (Orlikowski, 2002).</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Knowledge is an organized combination of data, assimilated with a set of rule, procedures, and operations learnt through experience and practice (Bhatt, 2001).</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Knowledge is contextual and includes an actionable summary and interpretation of experience (Amaravadi and Lee, 2005).</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Knowledge consists of the assimilation of related information addressed in the context of reference (Marsh, 1997).</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge can be defined as the situation where insight is achieved in a context by pointing out information from data as the difference that makes a difference (Jensen, 2005).</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge is information whose validity has been established through test of proof (Liebeskind, 1996).</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge is a clear understanding of information and their associated patterns (Bierly et al., 2000)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge is a collection of information and rules with which a certain function can be fulfilled (Hertog and Huizenga, 2000).</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge is meaningfully organized accumulation of information through experience, communication or inference (Bobrow and Collins, 1975).</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge is that which is known and so it is an essentially human form of information (Wilson, 1996).</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge consists of truths and beliefs, perspectives and concepts, judgments and expectations, methodologies and know-how (Wiig, 1993).</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Knowledge is a collective activity embodied in situated practice (Lave and Wenger, 1991).</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Information used to take decisions forms knowledge upon which people base actions to achieve results (Davenport et al., 2001)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Legend: A- Agent, C- Action, S-Current state of affairs, G- Goals and I- Information

Table 17: Analysis of definitions of knowledge in KM literature
Table 18: Summary of knowledge related concepts that appear in knowledge definitions (in Table 17)

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Total</th>
<th>With Agent</th>
<th>With Action</th>
<th>With State</th>
<th>With Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>9</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Action</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Goal</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 19: Summary of other concepts that appear in knowledge definitions (in Table 17)

<table>
<thead>
<tr>
<th>Number of times the concepts appear</th>
<th>Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ideas, facts, interpretations, communication, inference, functions, cognitive expectations, capacity, operations, methodology, perspectives, pattern, truth, accomplishment</td>
</tr>
<tr>
<td>2</td>
<td>Procedures, observations, judgment, expectations ability, know-how, concept, skill, and rule</td>
</tr>
<tr>
<td>4</td>
<td>Belief</td>
</tr>
<tr>
<td>7</td>
<td>Context, experience</td>
</tr>
</tbody>
</table>

Table 18: Summary of knowledge related concepts that appear in knowledge definitions (in Table 17)

<table>
<thead>
<tr>
<th>Number of times the concepts appear</th>
<th>Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ideas, facts, interpretations, communication, inference, functions, cognitive expectations, capacity, operations, methodology, perspectives, pattern, truth, accomplishment</td>
</tr>
<tr>
<td>2</td>
<td>Procedures, observations, judgment, expectations ability, know-how, concept, skill, and rule</td>
</tr>
<tr>
<td>4</td>
<td>Belief</td>
</tr>
<tr>
<td>7</td>
<td>Context, experience</td>
</tr>
</tbody>
</table>
Appendix B: Brief Description of OWL

This OWL overview is based on the official OWL documentation from the World Wide Web Consortium (W3C) (McGuinness et al., 2004). The key OWL constructs are classes, individuals and properties.

Classes in OWL are intended to represent concepts in a domain of discourse. They provide a mechanism for grouping resources with similar characteristics. Every OWL class can be associated with a set of individuals called the class extension. Two OWL class identifiers are predefined: owl:Thing and owl:Nothing. The class extension of the owl:Thing class is the set of all OWL individuals; thus, every OWL class is a subclass of owl:Thing. The class extension of owl:Nothing is the empty set; so owl:Nothing is a subclass of every class. The simplest way to define a class in OWL is just to declare it by name, for example:

```xml
<owl:Class rdf:ID="Student"/>
```

OWL classes are further defined through class descriptions. A class description describes an OWL class either by name (as shown above) or by specifying the class extension (set of instances) of an unnamed (anonymous) class. Defining classes in OWL by specifying the class extension means describing the conditions that must be satisfied by an individual for it to be a member of the class. A class in OWL can be described:

- By exhaustive enumeration of its individuals (using owl:oneOf construct for stating that the extension of a class consists of these and only these listed instances)
- As a set of all individuals which satisfy certain constraints on their properties (property restrictions).

OWL individuals represent objects in the domain of discourse. The individuals in the class extension are called instances of the class. Generally, it is intended that classes should correspond to naturally occurring sets of things in a domain of discourse and individuals should correspond to actual entities that can be grouped into these classes. For example, we can define a class Student with instances of this class (OWL individuals) representing some specific students. An individual can be minimally introduced by being declared a member of a class (either of the predefined top class owl:Thing or some other class defined in an ontology), for example:
In the above example, the first statement introduces an individual SomeThing simply as an instance of owl:Thing (no further information about this individual has been provided yet). The second statement declares another individual, John_Doe, which is stated to be an instance of the class Human (note that this individual is automatically an instance of owl:Thing since any OWL class is the subclass of owl:Thing).

All OWL properties are binary relationships. They are used to assert general facts about class and specific facts about individuals. There are two main types of properties:

- **Object properties** relate individuals to individuals. For example, in an ontology that describes humans, we can define an object property hasParent to relate individuals representing persons to other individuals - their parents.

- **Datatype properties** link individuals to data values (an XML schema datatype value or an RDF literal). For example, we may define a datatype property hasHeight to represent the height of a person, i.e. to link an individual (person) to a nonnegative integer representing height.

Properties in OWL have direction: a property links a subject (an OWL individual) to an object (an OWL individual or a data value), and the object is considered a value of this property for the subject. Properties in OWL are described using property axioms. In its simplest form, a property axiom just declares the existence of a property by its name, for example

```xml
<owl:ObjectProperty rdf:ID="hasParent"/>
```

Properties may have a domain and/or a range specified (using class descriptions and XMLS schema datatypes). For example, in some ontology we may need to specify that the domain of a property hasAge is the class Human and the range is a set of nonnegative integers (represented as an XML Schema datatype). This is represented as:

```xml
<owl:Class rdf:ID="Human">
<owl:DatatypeProperty rdf:ID="hasAge">
<rdfs:domain rdf:resource="#Human"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#nonnegativeInteger"/>
</owl:DatatypeProperty>
</owl:Class>
```
Appendix C: Specific Rules to Model Interactions

(Cited in page 97)

- Instances of interacting classes must have at least one mutual property which is modeled as the property of the interaction classes. In the absence of any mutual property, instances do not interact.

- Each interaction class must have at least two object properties (that can be identified with appropriate prefixes) linking it to the interacting class. This restriction reflects that at least two interacting classes are necessary to form one interaction class.

- Each interacting class must have at least one instance. By enforcing this restriction, it is made explicit that instances that interact with each other exist.

- Each interaction class represents a set of related concurrent mutual properties (arising out of the same interaction). Different interaction classes should be used if sets of properties are not concurrent.
Appendix D: Training materials used in the experiment in Chapter 4

In this experiment we will introduce to you a particular mechanism to represent information of a domain. The key concepts used in the mechanism are described next.

1. A **class** (represented by rectangle box) is a name of a concept from a domain. For example, in the domain of student registration, student, university, courses can be considered as classes.

2. A class is described by some attributes called **properties**. Properties are of two types. First, properties that describe the class itself (such as grade and name are the properties of the class student) and second, properties that relates one or more classes to one or more other classes (such as a property *take* can connect the students class with the courses class, indicating the statement students take courses). In the former type, the properties are mentioned within the class (such as property A1 and property A2 inside the class A) and in the latter type, the property is mentioned along with an arrow that connects the two classes (refer to Fig 36). The latter type of property is also mentioned inside the class from where the arrow originates. Any number of arrows (with the name of the property mentioned along with) can connect to any number of classes. It is to be noted that all the properties mentioned in this representation are mandatory. This means that if a class student has properties (inside the class or connected to other classes) *age*, *grade* and *name*, then at all circumstances this class has these three properties. In other words, a class **student** cannot have just *name* or *age* as properties.

3. A **subclass** is a class that describes a class more specifically. The properties mentioned in a class are automatically inherited by its subclass (thus shown in the subclasses). An arrow (with white solid head) originates from the class to another class indicating that the former is the subclass of the latter. A subclass can also be formed by adding property connecting from one class to another class.

These three points are represented in Figure 37.
Figure 37: Representation of the concepts

To understand further how to represent information using the mechanism we use a specific example. Consider the situation that a student holds a subscription of a library. Student, Subscription and Library are shown as three classes and their relationships are indicated by properties of these classes. A student can borrow books, thus the Book class is related to the Student class by a property -borrow_book. Note that this property is mentioned in the Student class as the arrow connecting the Student and the Book originates from the Student class. A graduate student is a student who has taken some graduate courses. Thus Graduate_student is a subclass of the class Student. All the properties of the Student class are inherited by the Graduate_Student class and shown in the Graduate_Student class. This information is shown in Figure 38. Though it is explicitly mentioned that library possesses books, but such statement can also be inferred indirectly from the figure from the following three statements.

- A library has subscription (Has_Subscription property that connects classes Library and Subscription)
- Subscription is for student (For_Student property that connects classes Subscription and Student)
- Student borrows book (Borrows_Book property connects classes Student and Book)
Figure 38: Representation of the Library Example

Please note that all the properties (including properties to connect classes) are mandatory in this type of diagrams. For example, if four properties are mentioned in the student class then this class will always have four properties (no more or no less). In other words just Student_Number and Borrows_Book properties cannot define a student class.

In this experiment you will be asked to answer two types of questions (Yes/No and open ended) by viewing similar figures. To familiarize you with the type of answers expected from you, you will do a practice question now.

Question: You are asked to develop a procedure (a set of rules) for renewing subscription of students. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure for renewing subscription of students. Provide as many answers as possible.

Possible answers

1. Where is the student record located?
2. What is the student number of the student applying for renewal?
3. How to check the past dues of the student applying for renewal?
4. Can a library card be renewed without checking the previous dues?
5. How to check the current student status of the student applying for renewal?
6. How to inform the student to pick up the renewed library card?

Though all of these answers are good, we would like you to focus on answers such as numbers 3-6. Asking these questions can help to form a procedure or a set of rules for renewing student subscriptions. Please note that there are some questions that can help to develop a procedure directly (such as 3, 5 and 6) where as there are some questions that help to develop the procedures indirectly (such as 4). For example, by asking question 3 —“how to check the past dues of the student?” a procedure can be developed as the students’ dues can
be checked in several ways such as by viewing computer record, or physically checking the books that the students ordered etc. and the procedure developed could be "check the student record in the computer and if the record is not found then physically check the books borrowed by the students."

When you are asked a similar question, your answers should be more like answers 3-6. You can refer to this example when you answer the questions.
Appendix E: Domain descriptions and modifications made in the ontologies for the experiment in Chapter 4

Following are the descriptions of the two domains from where ontologies will be developed. These ontologies are obtained from at www.schemaweb.info and the description of these ontologies are from Zou (2004).

Description of the travel domain

A travel agent helps customers to organize travel plans. Travel agents organize hotel bookings, airline tickets and entertainment tickets. A customer has travel preferences including departure time and return date. A travel agent proposes a travel itinerary to a customer. A travel itinerary is complete if it satisfies the following conditions:
- It includes a round-trip airline ticket and hotel rooms for every night between the arrival and departure dates
- There can only be one hotel, i.e., the customer never moves to another hotel
- The return date must be later than departure date and cannot be on the same day

The reservation is a contract between the customer and the service provider. Reservations are done by the travel agent when the travel itinerary is complete and the customer is satisfied with the itinerary. The customers select the travel agent who proposes the best travel itinerary. The customer’s preference is composed of the preference values for hotel, airline and entertainment.

Description of the auction domain

Auction is a competition-based method of allocating scarce resources. There are three participants in an auction. A seller, who owns the resources and wishes to obtain as much money as possible, holds an auction to extract information he might not otherwise realize. An auctioneer (auction house) acts as the agent for the seller. A buyer wants to pay as little as necessary. The two actors creating the auction are the seller who actually owns the goods and the auction house who shouts out for the seller. The Auction Item refers to the goods for auction. It may also be the combination of multiple auction items, for example, a travel package includes two airline tickets and one hotel room. Two important properties of the auction are closing time and reserved price. There are three statuses for the auction. The auction starts with “Open” status. When the auction is closed and items are sold, the status changes to “Close.” When the items are unsold, the status turns to “Failure.” The hasBidding property links to all submitted bids.

The bidder is the agent who participates in the auction and submits the bid. The bid has an expiration time, which is either a duration of time or the ending of the auction (which means never expire). The reserveResource refers to the resource reserved for the bid by the bidder. Note here that the auctionItem in the auction Class is the resource reserved for the auction by
the seller. For example, in a Priceline auction, the customer creates the auction stating the needed airline ticket and the suggested price. The airline company must reserves the related resource (air tickets) when submitting the bid and make sure the tickets are available when it wins the auction. Of course when the bidder loses the auction, the reserved resource will be immediately released. The reserved resource may also be other resources related to the auction items, for example, a reserved IP address when the bidder bids for a new computer, or the money on hold in the bank account when bidding for the auction item.

**Modifications of the ontologies**

The original ontologies, modified ontologies (uninformed) and informed ontologies are presented in Figures 39-44. Few changes were made to the original ontologies to obtain the uninformed ontologies after the equivalent informed ontologies were developed. The main reason for doing these changes is to make these ontologies informationally equivalent. In the auction ontology, two subclasses were added—WinningBidding and ItemsSold. These subclasses are then connected with two object properties. The object properties from Auction to Participant were renamed as hasAuctionHouse and hasSeller to make the meaning of these properties more meaningful. The object properties connecting Bidding and Auction and connecting Auction and AuctionItem were made bi-directional to make the information content in both the informed and uninformed ontology equivalent.

In the travel ontology two classes were introduced—Person and InitialItinerary. On consultation with a travel agent it was realized that itinerary is generally referred in two ways. When a person comes to the travel agent to explore the possibility of traveling. The print out of details of travel plan given at this stage is called Initial itinerary. Once the person is satisfied with the initial itinerary a final itinerary is printed when the person books the ticket. Creating two new classes—Person and FinalItinerary reflects this differentiation of itinerary. The original Itinerary class is renamed as InitialItinerary. To maintain informational equivalency between the informed ontology and the uninformed ontology two more object properties were added to the FinalItinerary and the ServiceProvider classes. The original ontology obtained from the web had several subclasses that contain the same properties as that of the main class. These are: EntertainmentPreference, AirlinePreference, HotelPreference, EntertainmentReservation, AirlineReservation and HotelReservation. These subclasses were removed from both the informed ontology and the uninformed ontology to decrease the complexity of the domain. All these changes were made keeping the overall structure of the original ontologies same.
Appendix F: Ontologies used in the experiment described in Chapter 4

Figure 39: Auction ontology originally obtained from the web
Figure 40: Modified auction ontology (uninformed)
Figure 41: Informed Auction ontology
Figure 42: Travel ontology originally obtained from the web
Figure 43: Modified travel ontology (uninformed)
Figure 44: Modified travel ontology (informed)
## Appendix G: Mapping Concepts between Ontologies

<table>
<thead>
<tr>
<th>Uninformed ontology</th>
<th>Informed Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Construct</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>InitialItinerary</td>
<td>Class</td>
</tr>
<tr>
<td>FinalItinerary</td>
<td>Class</td>
</tr>
<tr>
<td>Person</td>
<td>Class</td>
</tr>
<tr>
<td>Customer</td>
<td>Class</td>
</tr>
<tr>
<td>Preference</td>
<td>Class</td>
</tr>
<tr>
<td>Reservation</td>
<td>Class</td>
</tr>
<tr>
<td>ServiceProvider</td>
<td>Class</td>
</tr>
<tr>
<td>offerPrice</td>
<td>Property of InitialItinerary</td>
</tr>
<tr>
<td>Penaltyprice</td>
<td>Property of InitialItinerary</td>
</tr>
<tr>
<td>travelAgent</td>
<td>Property of InitialItinerary</td>
</tr>
<tr>
<td>forPerson</td>
<td>Property of InitialItinerary</td>
</tr>
<tr>
<td>returnDate</td>
<td>Property of InitialItinerary</td>
</tr>
<tr>
<td>departureDate</td>
<td>Property of InitialItinerary</td>
</tr>
<tr>
<td>to</td>
<td>Property of InitialItinerary</td>
</tr>
<tr>
<td>from</td>
<td>Property of InitialItinerary</td>
</tr>
<tr>
<td>offerPrice</td>
<td>Property of FinalItinerary</td>
</tr>
<tr>
<td>penaltyPrice</td>
<td>Property of FinalItinerary</td>
</tr>
<tr>
<td>travelAgent</td>
<td>Property of FinalItinerary</td>
</tr>
<tr>
<td>forCustomer</td>
<td>Object property of FinalItinerary</td>
</tr>
<tr>
<td>hasReservation</td>
<td>Object property of FinalItinerary</td>
</tr>
<tr>
<td>returnDate</td>
<td>Property of FinalItinerary</td>
</tr>
<tr>
<td>Departuredate</td>
<td>Property of FinalItinerary</td>
</tr>
<tr>
<td>To</td>
<td>Property of FinalItinerary</td>
</tr>
<tr>
<td>from</td>
<td>Property of FinalItinerary</td>
</tr>
<tr>
<td>name</td>
<td>Property of Person</td>
</tr>
<tr>
<td>Construct</td>
<td>Informed Ontology</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>hasPreference</td>
<td>Object property of Person</td>
</tr>
<tr>
<td>hasInitialItinerary</td>
<td>Object property of Person</td>
</tr>
<tr>
<td>travelPlanReady</td>
<td>Property of Person</td>
</tr>
<tr>
<td>PreferValue</td>
<td>Property of Preference</td>
</tr>
<tr>
<td>Name</td>
<td>Property of Customer</td>
</tr>
<tr>
<td>hasInitialItinerary</td>
<td>Object property of Customer</td>
</tr>
<tr>
<td>hasFinalItinerary</td>
<td>Object property of Customer</td>
</tr>
<tr>
<td>hasPreference</td>
<td>Object property of Customer</td>
</tr>
<tr>
<td>travelPlanReady</td>
<td>Property of Customer</td>
</tr>
<tr>
<td>Price</td>
<td>Property of Reservation</td>
</tr>
<tr>
<td>reserveNumber</td>
<td>Property of Reservation</td>
</tr>
<tr>
<td>byCustomer</td>
<td>Object property of Reservation</td>
</tr>
<tr>
<td>hasFinalItinerary</td>
<td>Object property of Reservation</td>
</tr>
<tr>
<td>hasServiceProvider</td>
<td>Object property of Reservation</td>
</tr>
<tr>
<td>serviceProviderName</td>
<td>Property of ServiceProvider</td>
</tr>
<tr>
<td>serviceType</td>
<td>Property of ServiceProvider</td>
</tr>
<tr>
<td>hasReservation</td>
<td>Object property of ServiceProvider</td>
</tr>
</tbody>
</table>

Table 20: Establishing information equivalency between the travel ontologies (from uninformed to informed)
<table>
<thead>
<tr>
<th>UnInformed Ontology</th>
<th>Informed Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Construct</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Bidding</td>
<td>Class</td>
</tr>
<tr>
<td>Auction</td>
<td>Class</td>
</tr>
<tr>
<td>WinningBidding</td>
<td>Class</td>
</tr>
<tr>
<td>Participant</td>
<td>Class</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>AuctionItem</td>
<td>Class</td>
</tr>
<tr>
<td>ItemSold</td>
<td>Class</td>
</tr>
<tr>
<td>bidder</td>
<td>Property of class Bidding</td>
</tr>
<tr>
<td>bidQuantity</td>
<td>Property of class Bidding</td>
</tr>
<tr>
<td>bidExpiryTime</td>
<td>Property of class Bidding</td>
</tr>
<tr>
<td>bidMoney</td>
<td>Property of class Bidding</td>
</tr>
<tr>
<td>forAuction</td>
<td>Object property of class Bidding</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>reservePrice</td>
<td>Property of class Auction</td>
</tr>
<tr>
<td>hasAuctionItem</td>
<td>Object property of class Auction</td>
</tr>
<tr>
<td>closeTime</td>
<td>Property of class Auction</td>
</tr>
<tr>
<td>AuctionType</td>
<td>Property of class Auction</td>
</tr>
<tr>
<td>hasBidding</td>
<td>Object property of class Auction</td>
</tr>
<tr>
<td>hasAuctionHouse</td>
<td>Object property</td>
</tr>
<tr>
<td>property</td>
<td>type</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>hasSeller</td>
<td>Object property of class Auction</td>
</tr>
<tr>
<td>hasItemSold</td>
<td>Object property of class WinningBidding</td>
</tr>
<tr>
<td>isWinningBidder</td>
<td>Property of class WinningBidding</td>
</tr>
<tr>
<td>Name</td>
<td>Property of class Participant</td>
</tr>
<tr>
<td>SuggestRetailPrice</td>
<td>Property of AuctionItem</td>
</tr>
<tr>
<td>ItemQuantity</td>
<td>Property of AuctionItem</td>
</tr>
<tr>
<td>ItemName</td>
<td>Property of AuctionItem</td>
</tr>
<tr>
<td>ItemQuality</td>
<td>Property of AuctionItem</td>
</tr>
<tr>
<td>ItemDescription</td>
<td>Property of AuctionItem</td>
</tr>
<tr>
<td>ForAuction</td>
<td>Object property of AuctionItem</td>
</tr>
<tr>
<td>soldToWinningBidding</td>
<td>Object property of ItemSold</td>
</tr>
</tbody>
</table>

Table 22: Establishing information equivalency between the auction ontologies (from uninformed to informed)

<table>
<thead>
<tr>
<th>Informed Ontology</th>
<th>Uninformed ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construct</td>
<td>Type</td>
</tr>
<tr>
<td>InvolvesAuctionHouse</td>
<td>Bidding</td>
</tr>
<tr>
<td>InvolvesAuctionItem</td>
<td>Bidding</td>
</tr>
</tbody>
</table>

Table 23: Establishing information equivalency between the auction ontologies (from informed to uninformed)
| Table 24: Establishing information equivalency between the auction ontologies (from informed to uninformed) |  |
|---|---|---|
| **in ServiceProvider** | **hasReservation and Reservation hasFinalItinerary** | **combination of object properties** |
| **participatesInReservation** | **Object property in Customer class** | **Customer hasFinalItinerary and FinalItinerary hasReservation** | **Inference through combination of object properties** |
| **involvesTravelAgent** | **Reservation** | **Reservation hasFinalItinerary and FinalItinerary has travelAgent** | **Inference through combination of object properties** |
Appendix H: Test Materials of the experiment in Chapter 4

Auction Domain

Comprehension Questions [Answers are true/false]
1. A bidder can contact a seller directly bypassing the auction house
2. An item is sold only to a winning bidder
3. An auction can be performed without involving the seller
4. A bidder must bid a specific quantity of the item for sale
5. An auction can be performed without involving the auction house
6. An auction house has a name
7. Every bidder is a winning bidder
8. Every item is sold to some bidder
9. An item must have a description

Knowledge identification Tasks
1. You are asked to develop a procedure (a set of rules) to allow canceling bids proposed by bidders. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure to allow retracting bids proposed by bidders. Provide as many responses as you can.
2. You are asked to develop a procedure (a set of rules) for stopping bidders to buy directly from sellers without the knowledge of auction house. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure to stop bidders buying directly from sellers without the knowledge of auction house. Provide as many responses as you can.
3. You are asked to develop a procedure (a set of rules) for preventing winning bidders not paying for the item that they have won. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure for preventing winning bidders not paying for the item that they have won. Provide as many responses as you can.

Sample answers for Knowledge identification Tasks used as feedback
Response for task 1
How to provide information to the seller/auction house about cancellation of bids?
How to provide immediate notice to the auction house?
How to get the money back to the bidder?
How to determine penalty (if any) for the bid cancellation?
How many times can one cancel a bid?
How to identify the bidder who wants to cancel the bid?
How to set up a limitation on type of products whose bids cannot be cancelled?
How to set up a time limit within which the cancellation must be carried out?

Response for task 2
How to avoid direct contact with the seller during the bidding process?
Can the bidder communicate with the seller?
How to provide incentives (or rewards) to sellers for letting the auction house know when a bidder directly contacts the seller for buying?
Does a seller submit items to the auction house before they are auctioned?
How (where) to find out that bidders directly contact the seller for buying?
How to penalize the seller for selling items directly to the buyer?
How to penalize the bidder for buying items directly from seller?

Response for task 3
What kind of contract should the bidder sign before participating in the auction?
How to inform the auction house of non-payment?
How to penalize the bidder for not buying items after winning?
How to mention that the auction house is not responsible for selling once the bid is won?
Can an account with financial info (e.g. credit card) be provided to the auction house before bidding?
How to inform sellers that a bidder has not paid for other auctions that they have won in the past?
How to create incentives for winning bidders to pay for the item?

Travel domain

Comprehension Questions [Answers are true/false]

1. Every final itinerary must have a reservation
2. A service provider is involved in preparing initial itineraries
3. Every person is able to make reservations
4. Preparing final itinerary involves service providers
5. A reservation can be performed without involving a travel agent
6. A travel agent is involved in preparing final itineraries
7. Every initial itinerary must have a reservation
8. Reservation can be made without service provider’s involvement
9. Every itinerary should include departure date and return date

Knowledge identification Tasks

1. You are asked to develop a procedure (a set of rules) for cancellation of a reservation of a customer. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure for canceling reservation of a customer. Provide as many responses as you can.
2. You are asked to develop a procedure (a set of rules) for allowing customers travel without having reservations. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure for allowing customers travel without having reservations. Provide as many responses as you can.
3. You are asked to develop a procedure (a set of rules) for allowing customers to change their reservations. Using the above diagram as guidance, please specify the questions you
will ask in order to develop a procedure for allowing to change customers’ reservations. Provide as many responses as you can.

**Sample answers for Knowledge identification Tasks used as feedback**

**Response for 1**
How to check the record of the customer who wants to cancel?
How to inform the customer about the penalty for cancellation?
How to pay the penalty price (if there is a penalty for cancellation)?
How late will a customer be able to cancel a reservation?
How to contact the service provider/travel agent to cancel reservation?
How the travel agent informs the service provider about cancellation?
How to refund the money to the customer from the travel agent?

**Response for 2**
How to inform the customers that all seats are reserved or not reserved?
Can a final itinerary printout be used as a substitute of a reserved ticket?
How do service provider deal with double booking?
How is the price assigned for customers who travel without reservations?
How does the customer pay when he/she travels without reservations?
How to provide (assure) customers’ preference are available?
How to involve (inform) a service provider in preparing a final itinerary?

**Response for 3**
How to inform the service provider/travel agent about the change?
How late will a customer be able to change a reservation?
How to check the original reservation, itinerary and customer information?
How to inform the customer that the reservation has been changed (or not changed)?
Is there a penalty to change reservation? If so, how can it be applied?
How to pay the penalty price (if there is a penalty) or additional amount for the change?
Whether to delete the current reservation before making the changes?
How should the final itinerary be changed according to the change in reservation?
Whether to issue another reservation number or keep the old reservation number?

**Items for modeling familiarity (7-point likert scale)**

1. To what extent do you know data modeling concepts (such as entities, classes and properties)
2. To what extent do you have experience in using data modeling concepts (such as entities, classes and properties)

**Items for domain familiarity (7-point likert scale)**

1. On the last two years, to what extent have you made travel reservations?
2. On the last two years, to what extent have you participated in auctions (including online auctions)?
3. To what extent do you have knowledge of reservation procedures (e.g., used by ticketing companies, airlines)?
4. To what extent do you have knowledge of auction procedures?

**Items for Perceived Ease of Use (7-point likert scale)**
1. To what extent is the information **represented** in the diagrams easy to understand?
2. To what extent is the information **represented** in the diagrams confusing?
3. Trying to understand all of the information **represented** in the diagram required a lot of mental effort?
4. Overall I found the information **represented** in the diagrams easy to interpret

**Items for Perceived Ease of Use (Syntax) (7-point likert scale)**
1. To what extent is the **arrangement** of items (e.g. classes, properties) in the diagrams easy to understand?
2. To what extent is the **arrangement** of items (e.g. classes, properties) in the diagrams clear?
3. Overall I found the **arrangement** of items (e.g. classes, properties) in the diagrams easy to interpret

**Items for Perceived Understanding (7-point likert scale)**
1. To what extent did you understand all of the information **represented** in the diagram?
2. To what extent did you comprehend all of the information **represented** in the diagram?
3. Overall I grasped all of the information **represented** in the diagram
Appendix J: Coding Manual For the study in Chapter 4

Introduction

This guide is to facilitate you in evaluating subject responses. The responses are in the form of questions that lead to some procedures. A procedure is a set of actions that is required to perform a task. Consider that a procedure is being developed for renewing students' library card. In this case, students' library card expires after a certain date and the students need to renew their library card. A procedure can be developed that may assist a librarian to renew students' library card. In other words, a librarian (who is approached by the student to renew his/her library card) can follow the procedure of renewing library card by looking at the list of actions mentioned in the procedure. Developing such procedure is helpful to an organization as a developed procedure can become a part of standard practices (as it is formalized) and ready to be used by anybody in the organization.

Task

Subjects in this study are requested to come up with questions that lead to develop procedures. A typical task is as follows:

You are asked to develop a procedure (a set of rules) for renewing subscription of students. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure for renewing subscription of students. Provide as many answers as possible.

Typical responses

The subjects use the database representation (diagram) to understand the domain (in this case renewing library card) and they come up with a set of questions such as:

1. Where is the student record located?
2. What is the student number of the student applying for renewal?
3. How to check the past dues of the student applying for renewal?
4. Can a library card be renewed without checking the previous dues?
5. How to check the current student status of the student applying for renewal?
6. How to inform the student to pick up the renewed library card?
Clarification of responses

Not all questions (as responses) will help to form procedures. A response such as “what is the student number?” will not be useful to create a procedure for renewing library card, as the answer of this question is a particular student number. Rather a question such as “how to locate the student record?” can be considered as a question that will help to form a procedure. This is because a student’s record can be located in several places (example- in physical files or in a database, etc.) and a question such as “how to locate a students’ record?” will help to create a set of actions as part of the procedure for renewing library membership. Similarly, if the following question is asked: “how to inform the students to pick up renewed library card?”, then the response to this question is a set of actions such as mail the library card, or if the student lives on the university campus then ask him/her to pick up the card from the circulation desk.

Out of all these questions (1-6), only questions 3 to 6 can help to form procedures, questions 1 and 2 do not help to form procedures.

Types of responses

Subject responses can be classified into three types. In the first type, no procedures can be formed (example – questions 1 and 2). In the second type, there are questions that will help to form procedures directly (such as 3, 5 and 6). For example, by asking question 3 –“how to check the past dues of the student?” a procedure for checking the past due dates can be developed, as the students’ dues can be checked in several ways such as by viewing computer record, or physically checking the books that the students ordered and the procedure developed could be “check the student records in the computer and if the record is not found then physically check the books borrowed by the students.” Therefore this is a question that leads directly to develop a procedure. The third type help to develop procedures indirectly (such as 4). Question such as “are students’ record also stored physically?” can be considered as an indirect question to develop procedure. This is because the answer to this question is Yes or No. If the answer is yes, then the action “check students record physically” stays as part of the set of actions for the procedure of checking the due date, otherwise if the answer is no, this action is deleted from the procedure. Example of another indirect question leading to the development of procedure is “To locate students’ record, do you need to check the computer database first or the physical records stored in the file first.” Answer to this question will help develop the sequence of actions for the procedure “checking students records.”

Domain description

Following is the description of the domains from which the database representations are developed. Based on the understanding of the domains, subjects develop questions that lead to form procedures.
Description of the travel domain

A travel agent helps customers to organize travel plans. Travel agents organize hotel bookings, airline tickets and entertainment tickets. A customer has travel preferences including departure time and return date. A travel agent proposes a travel itinerary to a person. Such itinerary is called initial itinerary. A travel itinerary is complete if the reservation is made. Such itinerary is called the final itinerary. The reservation is a contract between the customer and the service provider. The travel agent does reservations when the final itinerary is complete and the customer is satisfied with the itinerary. The customer’s preference is composed of the preference values for hotel, airline and entertainment.

Description of the auction ontology

There are three participants in an auction- a seller, who owns the resources, an auctioneer (auction house) who acts as the agent for the seller and a bidder. The Auction Item refers to the goods for auction. Two important properties of the auction are closing time and reserved price. The bidder is an agent who participates in the auction and submits a bid. The bid has an expiration time. On winning a bid, a bidder becomes a buyer (or winning bidder). The seller sells the winning bidder the item that the bidder has won.

Domain based tasks

Travel Domain

1. You are asked to develop a procedure (a set of processes) for cancellation of a reservation of a customer. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure for canceling reservation of a customer. Provide as many responses as you can.

2. You are asked to develop a procedure (a set of processes) for allowing customers travel without having reservations. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure for allowing customers travel without having reservations. Provide as many responses as you can.

3. You are asked to develop a procedure (a set of processes) for allowing customers to change their reservations. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure for allowing to change customers’ reservations. Provide as many responses as you can.
Auction Domain

1. You are asked to develop a procedure (a set of processes) to allow canceling bids proposed by bidders. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure to allow retracting bids proposed by bidders. Provide as many responses as you can.

2. You are asked to develop a procedure (a set of processes) for stopping bidders to buy directly from sellers without the knowledge of auction house. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure to stop bidders buying directly from sellers without the knowledge of auction house. Provide as many responses as you can.

3. You are asked to develop a procedure (a set of processes) for preventing winning bidders not paying for the item that they have won. Using the above diagram as guidance, please specify the questions you will ask in order to develop a procedure for preventing winning bidders not paying for the item that they have won. Provide as many responses as you can.

Sample answers for the tasks

It is to be noted that other correct answers are possible

Travel domain

Response for 1
How to check the record of the customer who wants to cancel?
How to inform the customer about the penalty for cancellation?
How to pay the penalty price (if there is a penalty for cancellation)?
How late will a customer be able to cancel a reservation?
How to contact the service provider/travel agent to cancel reservation?
How the travel agent informs the service provider about cancellation?
How to refund the money to the customer from the travel agent?

Response for 2
How to inform the customers that all seats are reserved or not reserved?
Can a final itinerary printout be used as a substitute of a reserved ticket?
How do service provider deal with double booking?
How is the price assigned for customers who travel without reservations?
How does the customer pay when he/she travels without reservations?
How to provide (assure) customers’ preference are available?
How to involve (inform) a service provider in preparing a final itinerary?

Response for 3
How to inform the service provider/travel agent about the change?
165

How late will a customer be able to change a reservation?
How to check the original reservation, itinerary and customer information?
How to inform the customer that the reservation has been changed (or not changed)?
Is there a penalty to change reservation? If so, how can it be applied?
How to pay the penalty price (if there is a penalty) or additional amount for the change?
Whether to delete the current reservation before making the changes?
How should the final itinerary be changed according to the change in reservation?
Whether to issue another reservation number or keep the old reservation number?

**Response domain**

**Response for 1**

How to provide information to the seller/auction house about cancellation of bids?
How to provide immediate notice to the auction house?
How to get the money back to the bidder?
How to determine penalty (if any) for the bid cancellation?
How many times can one cancel a bid?
How to identify the bidder who wants to cancel the bid?
How to set up a limitation on type of products whose bids cannot be cancelled?
How to set up a time limit within which the cancellation must be carried out?

**Response for 2**

How to avoid direct contact with the seller during the bidding process?
Can the bidder communicate with the seller?
How to provide incentives (or rewards) to sellers for letting the auction house know when a bidder directly contacts the seller for buying?
Does a seller submit items to the auction house before they are auctioned?
How (where) to find out that bidders directly contact the seller for buying?
How to penalize the seller for selling items directly to the buyer?
How to penalize the bidder for buying items directly from seller?

**Response for 3**

What kind of contract should the bidder sign before participating in the auction?
How to inform the auction house of non-payment?
How to penalize the bidder for not buying items after winning?
How to mention that the auction house is not responsible for selling once the bid is won?
Can an account with financial info (e.g. credit card) be provided to the auction house before bidding?
How to inform sellers that a bidder has not paid for other auctions that they have won in the past?
How to create incentives for winning bidders to pay for the item?

**Filling up the code sheet**

Two code sheets (for two domains) are provided to evaluate the responses. You need to evaluate each response provided by the subject. The first three columns in the code sheet
will help you to determine whether the response is correct or incorrect. In the first column, you need to indicate whether a procedure can be developed or not from the response. Indicate Yes or No in this column. The next question asks whether the response helps to develop procedure for the particular task or not. For example, a response such as “how to refund the money from the service provider to the travel agent?” can help to form a procedure but is not related to the procedure for canceling reservation of a customer. In the third column you have to indicate whether the response helps to create procedures directly or indirectly. Answering these questions will help you to determine the score. Indicate 0 if the response does not lead to developing the given procedure, otherwise indicate 1. Finally in the last column, indicate your confidence level for each response (on a scale of 1 to 3, where 1-Low and 3-High). Rarely a subject may come up with more than 10 responses for a particular task. In that case use a new code sheet to mark the additional responses.
### Appendix K: Analysis of the terms used in the ontologies

<table>
<thead>
<tr>
<th>Terms</th>
<th>Response Subject 1</th>
<th>Response Subject 2</th>
<th>Response Subject 3</th>
<th>Response Subject 4</th>
<th>Standard definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Itinerary</td>
<td>Detailed schedule of traveling plan</td>
<td>Trip plan; detailed description of a trip</td>
<td>The description of route leading to a destination</td>
<td>It regards the route that a person plans for a trip, visit, etc.</td>
<td>The route of a journey or tour or the proposed outline of one</td>
</tr>
<tr>
<td>Customer</td>
<td>Someone who purchased products/services from the provider</td>
<td>Client; person who purchases a service/product</td>
<td>Firms or individuals who buy products or services</td>
<td>It is a wide concept that refers to people who get rather than provide a service.</td>
<td>One that purchases a commodity or service</td>
</tr>
<tr>
<td>Preference</td>
<td>Taste, favor something over others</td>
<td>Taste; what someone wants out of several options</td>
<td>The evaluation for a product or service</td>
<td>The first choices among all other alternatives</td>
<td>The act, fact, or principle of giving advantages to some over others</td>
</tr>
<tr>
<td>Reservation</td>
<td>Book in advance</td>
<td>Booking of a service</td>
<td>Booking a space before consumption</td>
<td>Holding for a period of time; book</td>
<td>An arrangement to have something (as a hotel room) held for one's use</td>
</tr>
<tr>
<td>Service Provider</td>
<td>Someone who provides certain services</td>
<td>Individual/company offering a service (the way I see it, usually Internet based)</td>
<td>Firms or individuals who provide the service</td>
<td>Those who provide services</td>
<td>A person who offers a service</td>
</tr>
<tr>
<td>Travel Agent</td>
<td>Person or company that help buyer find product relevant to traveling such as airline ticket and hotel room</td>
<td>Intermediary offering travel plan arrangements directly to clients</td>
<td>The middleman who facilitates the buying process of travel service</td>
<td>People who make a business on travel, e.g., trip planning, air-ticket reservation</td>
<td>A person engaged in selling and arranging transportation, tours, or trips for travelers</td>
</tr>
<tr>
<td>Travel Plan</td>
<td>A schedule for travel</td>
<td>Customer’s set of preferences relating to a trip</td>
<td>The guidelines to a trip, including ports</td>
<td>The plan for a trip regarding accommodation</td>
<td>A proposed route of travel</td>
</tr>
<tr>
<td>Price</td>
<td>(travel)</td>
<td>to visit, itinerary, time schedule, etc</td>
<td>itinerary, vehicle, etc.</td>
<td>The quantity of one thing that is exchanged or demanded in barter or sale for another</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>----------------------------------------</td>
<td>--------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Bidding</td>
<td>Money to be paid for a unit of product or service</td>
<td>Amount paid by customer for product/service</td>
<td>The money to be paid in order to exchange a product or service</td>
<td>The value asked for a merchandise</td>
<td></td>
</tr>
<tr>
<td>Bidding</td>
<td>Claim a price that the bidder could offer to buy something</td>
<td>To place an offer (on a product or service) in an auction setting</td>
<td>Submitting one’s willingness to pay for a product or service</td>
<td>To provide a price for a merchandise</td>
<td></td>
</tr>
<tr>
<td>Auction</td>
<td>Buyer and seller make transactions through bidding.</td>
<td>Selling mechanism by which interested buyers oppose each other by placing bids to purchase the item. The item is sold to one bidder in accordance to the rules of the auction.</td>
<td>A format of buying, where the final buyer is the one who are willing to pay most</td>
<td>To offer (a price) whether for payment or acceptance</td>
<td></td>
</tr>
<tr>
<td>Participant</td>
<td>Someone who participates (in an auction setting) persons taking part of an auction (i.e. players)</td>
<td>People involved in an activity</td>
<td>Everyone who is part of an activity</td>
<td>One that participates</td>
<td></td>
</tr>
<tr>
<td>Auction Item</td>
<td>The item for auction</td>
<td>Product or service being sold by means of an auction</td>
<td>The object that participants are bidding for in an auction</td>
<td>The item that is being auctioned</td>
<td></td>
</tr>
<tr>
<td>Bidder</td>
<td>Someone who bids for some item in auction</td>
<td>Participant who places bids on the item being auction to hopefully acquire it</td>
<td>People who submit prices in order to win an auction item</td>
<td>Those who provide a price for a merchandise</td>
<td></td>
</tr>
<tr>
<td>Reserve Price</td>
<td>Lowest price the selling is willing to sell a product</td>
<td>A bidder’s maximum price he/she is willing</td>
<td>The minimum price that the seller is</td>
<td>The lowest price reserved by the part who</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A price announced at an auction as</td>
<td></td>
</tr>
<tr>
<td>Seller</td>
<td>Someone who want to sell something</td>
<td>Participant (of an auction) who wishes to sell an item</td>
<td>Firms or individuals who offer a product or service in exchange of money</td>
<td>Those who sell an item</td>
<td>One that offers for sale</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------</td>
<td>-------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Auction House</td>
<td>The place where the auction takes place</td>
<td>Participant supervising the auction process by putting on sell the item received by the seller and receiving bids from the bidders. The auction house decides to which bidder the item goes to in accordance to the rules of the auction.</td>
<td>The place that an auction is conducted</td>
<td>The place where auctions take place officially</td>
<td>An agent who sells goods at auction</td>
</tr>
</tbody>
</table>

Table 25: Analysis of the terms used in the ontologies