8-1-2010

A Novel Approach to Ontology Management

Jong Woo Kim
Georgia State University

PERMISSION TO BORROW

NOTICE TO BORROWERS

All dissertations deposited in the Georgia State University Library must be used only in accordance with the stipulations prescribed by the author in the preceding statement.

The author of this dissertation is:

Jong Woo Kim
3253 O Flowers Rd S
Atlanta, GA 30341

The director of this dissertation is:

Dr. Balasubramaniam Ramesh
Department of Computer Information Systems
J. Mack Robinson College of Business
Georgia State University
Atlanta, GA 30303

A NOVEL APPROACH TO ONTOLOGY MANAGEMENT

- WITH FOCUS ON CREATION, REFINEMENT, AND USE IN INFORMATION SYSTEMS

BY

JONG WOO (JONATHAN) KIM

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree

Of

Doctor of Philosophy

In the Robinson College of Business

Of

Georgia State University

GEORGIA STATE UNIVERSITY
ROBINSON COLLEGE OF BUSINESS
2010

ACCEPTANCE

This dissertation was prepared under the direction of Jong Woo Kim's Dissertation Committee. It has been approved and accepted by all members of that committee, and it has been accepted in partial fulfillment of the requirements for the degree of Doctoral of Philosophy in Business Administration in the Robinson College of Business of Georgia State University.

Dean: H. Fenwick Huss

DISSERTATION COMMITTEE

Dr. Balasubramaniam Ramesh (Chair)
Dr. Veda Storey
Dr. Mark Keil
Dr. Julia Hilliard

ACKNOWLEDGEMENTS

I would like to thank my dissertation committee for their guidance, encouragement, and support throughout my PhD program. I am grateful to my dissertation advisor, Dr. Ramesh, for his guidance, encouragement, and patience during all the phases of the research. Without his help and support, my completion of the PhD program would not be possible. Veda Storey has guided me to ontology research. She has provided me with the right directions for solving the problems that I faced during my research. Mark Keil gave me an invaluable research opportunity and support, through which I learned a lot about conducting quantitative research. I appreciate the opportunities to work and research with Julia Hilliard at the National Immunology Center. They provided me with financial stability and valuable interdisciplinary research opportunity. She has always kindly motivated me to work and study harder.

I will never forget the great academic experience during my PhD program at GSU. I thank the following faculty who offered invaluable doctoral seminar courses – Detmar Straub, Dan Robey, Lars Mathiassen, Mark Keil, Veda Storey, Vijay Vaishnavi, Phillip Gagne, and Satish Nargundkar. I thank my fellow PhD students for their friendship and support. I appreciate all the support from the CIS faculty and staff.

I am indebted to Choong C. Lee at Yonsei University, JeongShin Shin at Sunchon National University, JinGu Lee, HwangJick Ryu, JeYoung Ko at KOBACO, who encouraged me to pursue a PhD study in IS field.

I appreciate all the support and encouragement my parents, Sikyung Kim and Hooja Lee, have provided. Their lives of diligence and loyalty have taught me how to live my life. I am deeply thankful to my parents-in-laws, Youngsu Ahn and Myunghee Kim, for their support and trust. I especially appreciate my mother-in-law's prayer.

My family has always supported and trusted me throughout my academic journey in US. Without their sacrifice, I would never be able to complete my PhD study. I am always grateful to my beloved wife, Mesook Ahn, for her support. My children, Susie and Jiwon, have been very patient with me. Daddy will keep his promise to take you to fishing, swimming, and horseback riding.

Table of Contents

A NOVEL APPROACH TO ONTOLOGY MANAGEMENT
- WITH FOCUS ON CREATION, REFINEMENT, AND USE IN INFORMATION SYSTEMS

BY
JONG WOO (JONATHAN) KIM

Dissertation
Department of Computer Information Systems
J. Mack Robinson College of Business
Georgia State University
2010

Abstract

The term ontology is defined as the explicit specification of a conceptualization. While much of the prior research has focused on technical aspects of ontology management, little attention has been paid to the investigation of issues that limit the widespread use of ontologies and the evaluation of the effectiveness of ontologies in improving task performance. This dissertation addresses this void through the development of approaches to ontology creation, refinement, and evaluation.

This study follows a multi-paper model focusing on ontology creation, refinement, and its evaluation. The first study develops and evaluates a method for ontology creation using knowledge available on the Web. The second study develops a methodology for ontology refinement through pruning and empirically evaluates the effectiveness of this method. The third study investigates the impact of an ontology in use case modeling, which is a complex, knowledge intensive organizational task in the context of IS development. The three studies follow the design science research approach, and each builds and evaluates IT artifacts. These studies contribute to knowledge by developing solutions to three important issues in the effective development and use of ontologies.

Chapter 1

Introduction

Motivation

The term ontology was originally used to define a philosophical discipline. Today, ontology is used in a wide variety of disciplines, including computing. In the field of information systems, the term ontology is defined as the explicit specification of a conceptualization (Gruber 1993). Ontology can support a variety of applications including knowledge engineering, artificial intelligence, information retrieval, and integration of databases (Noy and McGuinness 2001).

The study and use of ontologies has been gaining attention over the past decade. Much of the prior research has focused on technical aspects of ontology management. This includes the development of languages for representing ontology (e.g. OWL, RDF) and ontology engineering tools (e.g. Protégé). However, not much attention has been paid to the investigation of issues that limit the widespread use of ontologies and the evaluation of the effectiveness of ontologies in improving task performance.

Several issues that limit the widespread use of ontologies have been identified (Peterson et al. 1998; Pinto and Martins 2004). *First*, ontology creation takes a lot of effort and time (Maedche and Staab 2000). This is one of the main obstacles facing the developers of ontologies. Especially, identifying a relevant knowledge source and organizing it as a part of an ontology are serious challenges (Tijerino et al. 2005). *Second*, once developed, ontologies become large, and their effective use is impeded by their complexity. Mechanisms to reduce the complexity of available ontologies will help improve their effective use (Conesa and Olivé 2004; Maedche and Staab 2001). Third, very few theoretically and empirically grounded studies on the effectiveness of ontologies in improving task performance have been conducted. The lack of such studies is an impediment to their wide adoption in practice (Gangemi et al. 2005). Motivated by the need to address these challenges, this research investigates the following research questions:

1. How can ontologies be created with minimal effort from widely available knowledge sources?

2. How can relevant aspects of knowledge be extracted from large ontologies to reduce the effort involved in their effective use?

3. How does the use of ontologies increase user satisfaction in complex knowledge intensive organizational tasks such as IS development?

The first question is addressed by the development and evaluation of a method for ontology creation using knowledge available on the Web. The second question is addressed by empirically evaluating the effectiveness of methods for pruning ontologies. The third question is addressed by investigating the impact of an interactive use of ontology on user satisfaction in a complex, knowledge intensive organizational task within the context of IS development. Specifically, I investigate whether an interactive use of an ontology improves user satisfaction in the retrieval of use cases during systems development. Each of these studies includes empirical investigations that are grounded in the following theories: cognitive fit (Vessey 1991), cognitive load (Sweller 1988; Sweller and Chandler 1994), and human-computer interaction (Sengupta and Te'eni 1993).

Relevant Literature

In this section, we discuss the extant literature that informs the three studies.

Ontology Creation

Ontology creation is still rather a craft than an engineering task (Pinto and Martins 2004). Existing ontology creation methodologies provide an array of options, techniques, and guidelines to help ontology construction (Corcho et al. 2003). Cristani and Cuel (2005) classify ontology creation methodologies as top-down and bottom-up. Top-down methods start with an abstract view of the domain and expand it with detailed specifications (e.g. KACTUS (Schreiber et al. 1995), DOLCE (Gangemi et al. 2002)). Bottom-up methodologies start from the specification of a certain task and obtain generalizations (e.g. TOVE (Gruninger and Fox 1995), OTK (Fensel et al. 2000)). Some methods take a middle-out method where the ontology creation starts with key concepts and then generalizations and specializations are created (e.g. Enterprise (Uschold et al. 1996), METHONTOLOGY (Fernández-López et al. 1997)).

Two major challenges shared by all these ontology creation methodologies are a) identifying a relevant knowledge source, and b) the significant (manual) effort involved in ontology creation. Much prior work has relied on the manual construction of ontologies from domain specific knowledge sources, which has proven to be extremely expensive (Lenat 1995). Therefore, there is increasing interest in the automated creation of ontologies from widely available knowledge sources. TANGO (Tijerino et al. 2003) and Text-To-Onto systems (Maedche and Volz 2001) are exemplars of such research. However, the World Wide Web which is a virtually infinite source of knowledge for almost any imaginable domain has been largely untapped for this purpose. If this knowledge could be extracted and organized, it could be effectively used to create domain ontologies. However, prior research on systematically analyzing and using the World Wide Web as a source of knowledge for the creation of domain ontologies is scant. Our research attempts to fill this gap by providing a methodology that is integrated with a well-established ontology creation approach, namely, METHONTOLOGY (Fernández-López et al. 1997).

While much of the prior research has focused on the development of methodologies for creating ontologies, the evaluation of the created ontologies has been limited to the identification of quality metrics for ontology and the evaluation of the quality of ontology (Burton-Jones et al. 2005; Guarino 2004). This research fills this gap by using the cognitive fit theory to investigate how the proposed ontology creation methodology can help develop better quality ontologies when compared with other approaches.

Ontology Refinement through Pruning

When an ontology becomes very large, it may no longer support the original purposes for which it was developed, because it is very difficult to find the relevant components of knowledge from it. For example, consider the Cyc ontology which is a huge commercial knowledge repository that was developed to capture and represent common sense knowledge. It contains more than 2.2 million assertions (facts and rules) describing more than 250,000 terms, including nearly 15,000 predicates. When queried with keywords, Cyc may provide a large amount of knowledge which often includes hundreds of irrelevant terms. Thus, collecting conceptually consumable information from large ontologies has proven to be a very difficult task. This makes it impossible to automate any process for using ontologies without using heuristics to infer their semantics and/or discard information that is irrelevant for a particular context.

Ontology pruning which removes irrelevant concepts is an effective method to retrieve relevant knowledge from large ontologies (Volz et al. 2003). A generic pruning task consists of two phases: selection phase and pruning phase (Conesa and Olivé 2006). The selection phase identifies elements relevant to the goals of the final ontology. The pruning phase deletes irrelevant or useless elements to create the ontology which has only relevant elements. Several ontology pruning methods (Studer et al. 1998; Swartout et al. 1996) support these two phases.

Prior research indicates that pruning is effective in specific domains medicine or defense (Studer et al. 1998; Swartout et al. 1996). While many authors claim that pruning increases the effectiveness of an ontology by retaining only relevant concepts (Volz et al. 2003), these claims have not been adequately examined and explained using a theoretically grounded empirical studies. To address this issue, this research uses the cognitive load theory to empirically investigate whether the approach to pruning developed in this research reduces cognitive load, and thus improves task performance.

Ontology Use in IS development

The third study in this research focuses on the use of a pruned ontology in the context of a problem which requires rich semantic knowledge provided by the ontology. Specifically, we develop and evaluate a methodology for the use of ontologies in *use case modeling* in software development.

A *use case* is a key artifact that is created and managed throughout the entire processes of system development (Jacobson 1992). The creation of use cases is often the first step in the acquisition of requirements from users. It is an effective communication vehicle to capture requirements from users. Other design artifacts such as state transition diagrams and class diagrams are created based on use cases. Thus, use cases often represent a critical starting point in the system development life cycle. When stakeholders need to examine the relationships between the actual implementation and system requirements, they rely on use cases that document requirements.

A common task in requirements engineering involves the search for and the exploration of requirements which were created in earlier phases of a project or in other similar projects. Current requirements engineering tools support this function to a limited extent. They typically provide keyword based search capabilities. The ambiguity inherent in natural language usually limits the usefulness of such search (Sutton 2000). The objective of our research is to improve the retrieval of use cases from a library with the help of a relevant domain ontology..

Methodology

Study One

This study develops a six-step methodology shown in Figure 1 for semi-automatically generating domain ontologies from information available on the World Wide Web. Prior ontology creation methodologies are analyzed to develop the critical steps in this methodology. The methodology includes the following steps: 1) identification of the target domains, 2) specification of relevant web sites, 3) scanning information from the websites, 4) extraction of important concepts from the relevant web pages, 5) analysis of the extracted concepts, and 6) the construction of the ontology using these concepts.

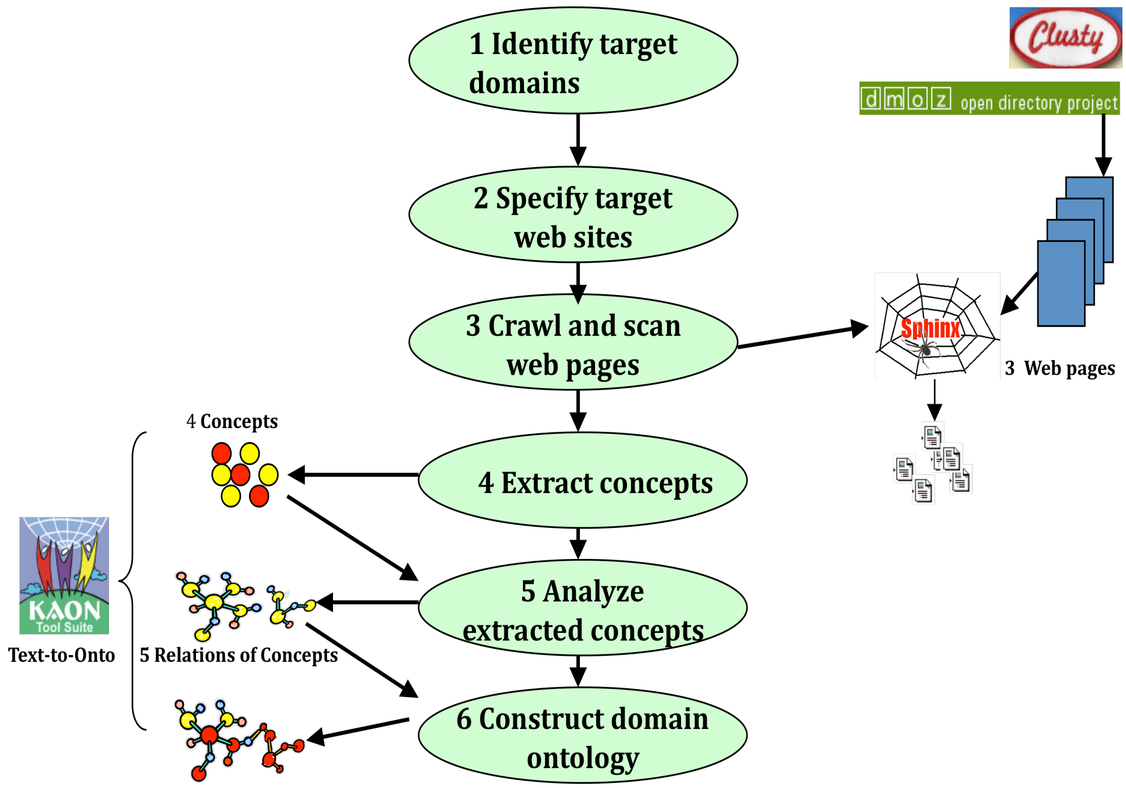


Figure 1. Six-step methodology for Automated Domain Ontology Creation

The methodology is implemented in a prototype that creatively combines and refines partial solutions for each step. A prototype called WebtoOnto created in this research comprises of three modules shown in Figure 2. It is used to develop ontologies for various application domains. An empirical analysis that uses the cognitive fit theory theory (Vessey 1991) is carried out to demonstrate the feasibility of the proposed methodology. The research model used in the evaluation (shown in Figure 3) suggests that decision makers can deliver faster and more accurate solutions when the presented information matches the mental representation of their decision task. Five hypotheses that were developed based on the theory are tested using an experiment. The use of WebtoOnto provides cognitive fit with the task of creating domain ontologies. In the experimental evaluation, the performance of subjects who create domain ontologies with and without cognitive fit are compared.

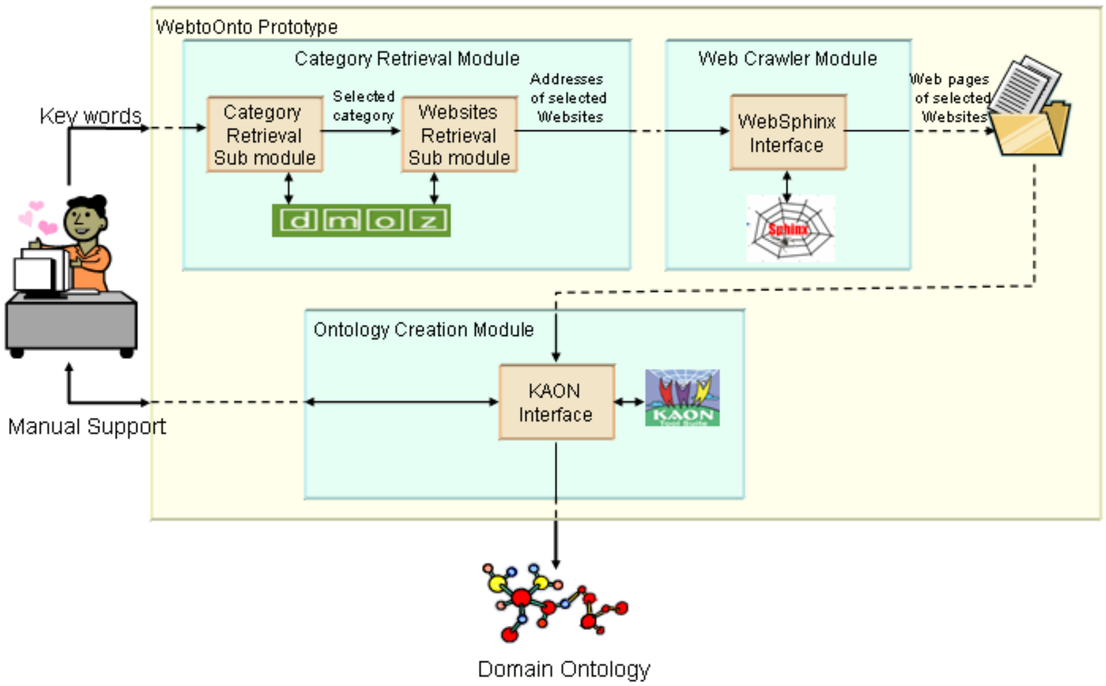


Figure 2: System Architecture of WebtoOnto – Study One

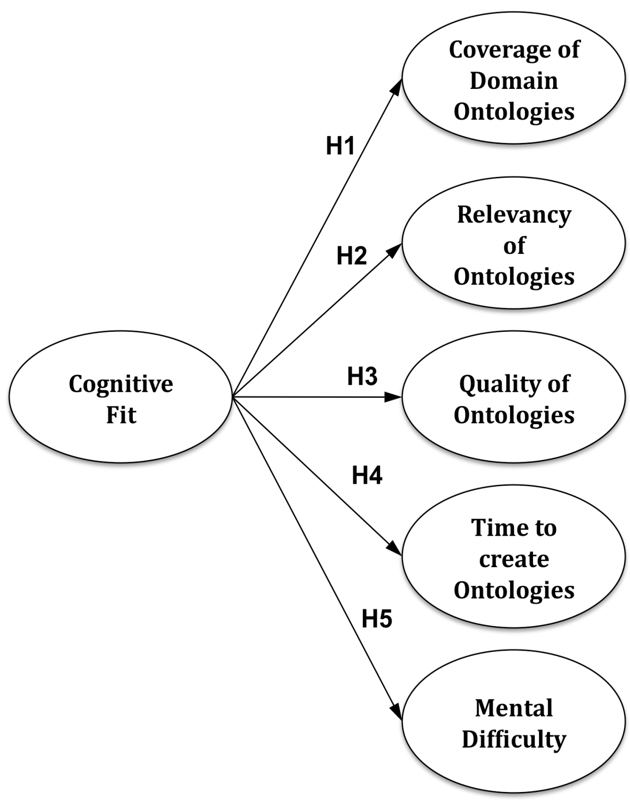


Figure 3: Research Model – Study One

Study Two

The second study focuses on ontology pruning and examines how it influences task performance in a complex domain, viz., the use of a complex ontology used in biology. A prototype called GOP (Gene Ontology Pruner) is developed by significantly extending a pruning method developed in prior research. The architecture of the prototype is shown in Figure 4. It supports the systematic identification of concepts that are considered relevant and the deletion of irrelevant parts of an ontology. A large bio-ontology called Gene Ontology (GO) (Lee et al. 2006) is pruned to obtain a sub-ontology that contains only information that is of interest to the user.

The research model (shown in Figure 5) used in the empirical evaluation of the pruning method developed in this study is drawn from the cognitive load theory. This theory suggests that ontology pruning, which is a form of information filtering, reduces cognitive load. The model suggest that cognitive load, in turn, affects task performance.

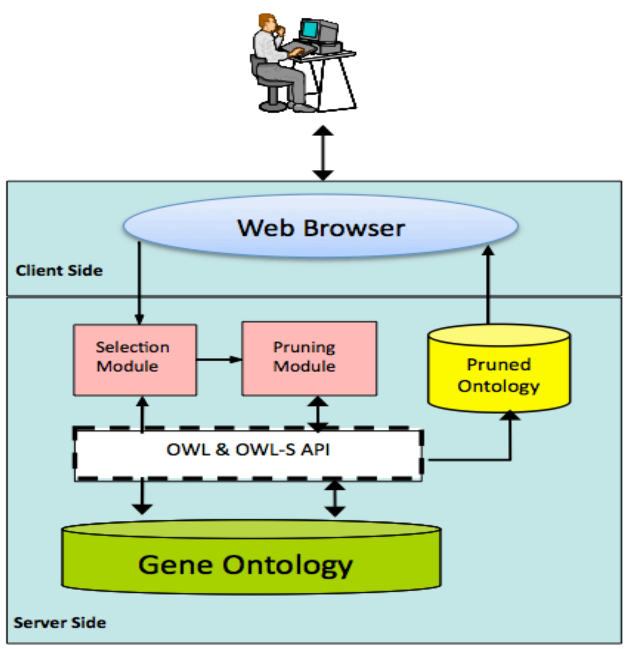


Figure 4: System Architecture of GOP – Study Two

The ontology pruning method is evaluated using an experimental study. In this experiment, the performance of subjects with access to unpruned and pruned ontologies is compared. The effects of cognitive fit and task complexity on cognitive load are also examined. The effects of cognitive load on task performance quality and efficiency are studied. Quality is measured in terms of the accuracy of the answers to questions on ontological knowledge and efficiency is measured in terms of the time taken to complete the task.

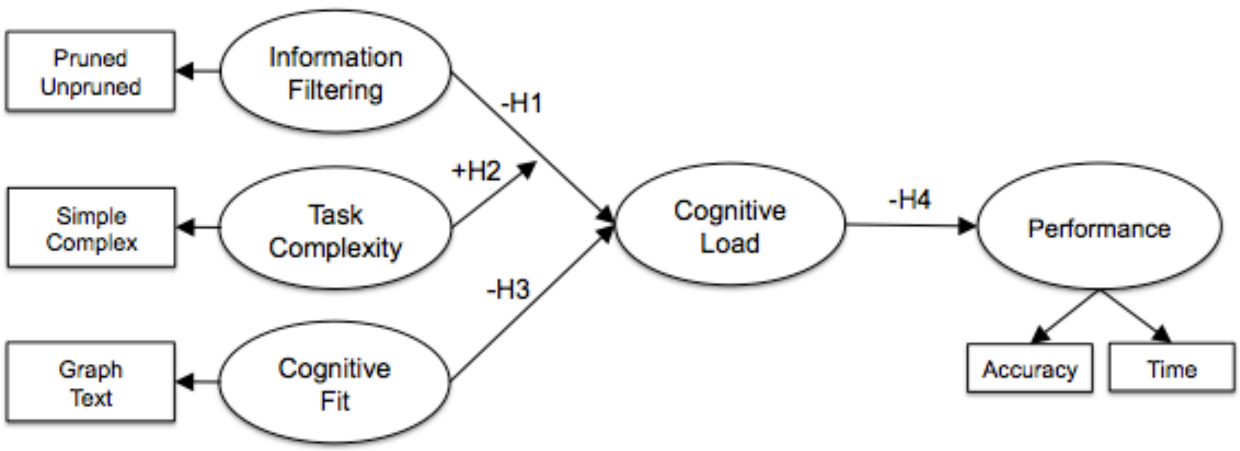


Figure 5: Research Model – Study Two

Study Three

The focus of the last study is the use of ontology to increase user satisfaction in retrieving use cases from a repository. Use cases are popular because of the use of natural language, which however poses interesting challenges. Use cases expressed in natural language are likely to be inherently imprecise, ambiguous, incomplete and inconsistent. Present case tools provide only keyword based search capability to retrieve use cases. Our work is based on the premise that the reuse of use cases can be improved by taking advantage of the semantic knowledge embedded in ontologies. Motivated by this premise, our research uses an ontology approach to accurately retrieve use cases.

The challenge facing this ontological approach is to select and suggest relevant use cases based on a query, which is interactively developed by a user. Figure 6 shows the architecture of the proposed system that has four modules: Query Parser Module, Concept Identification Module, Inference Module, and Interface Module. The interface module enables the interaction between users and the system. The query parser module receives the user's query from interface module and parses it to return the part-of-speech for each term. The concept identification module interacts with ontologies to retrieve relevant concepts that are presented to the user via the interface module. The user interacts with the system by selecting the concepts of interest to him/her. The inference module receives the selected concepts and finds relevant use cases.

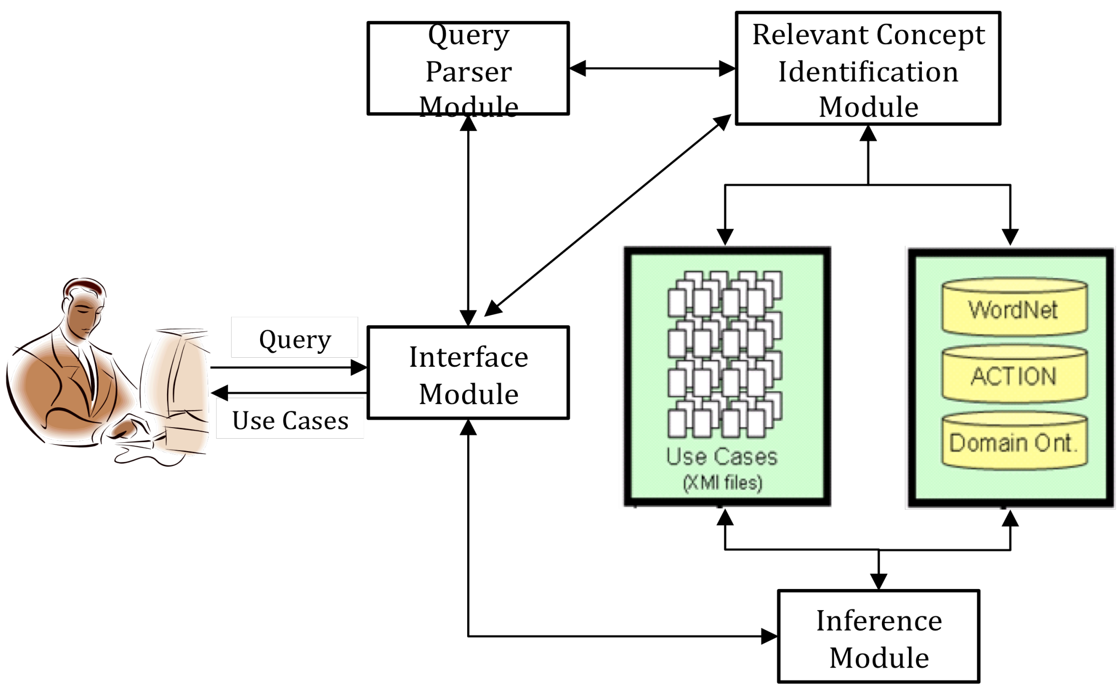


Figure 6: System Architecture – Study Three

A research model used in the experimental evaluation of the effectiveness of this approach is shown in Figure 7. This study investigates how information filtering and interaction affect cognitive load and self-efficacy, which, in turn, affect user satisfaction. Interaction theory suggests that interaction between user and information systems supported by filtered information increases decision quality and confidence in individual decision making while decreasing cognitive load. When the user provides general and ambiguous terms as an input to the system, the system may not accurately capture the intended meaning. As a result, it generates irrelevant results. The system supported by ontologies can help the user refine his/her query by suggesting relevant concepts from ontologies. Through this interaction with the system, the user may feel that he/she controls how the system works. In addition the user may experience more satisfaction.

The research model is evaluated by laboratory experimentation. A 2 x 2 factorial design with information filtering and interaction is used to assess their impact on cognitive load, self-efficacy and satisfaction.

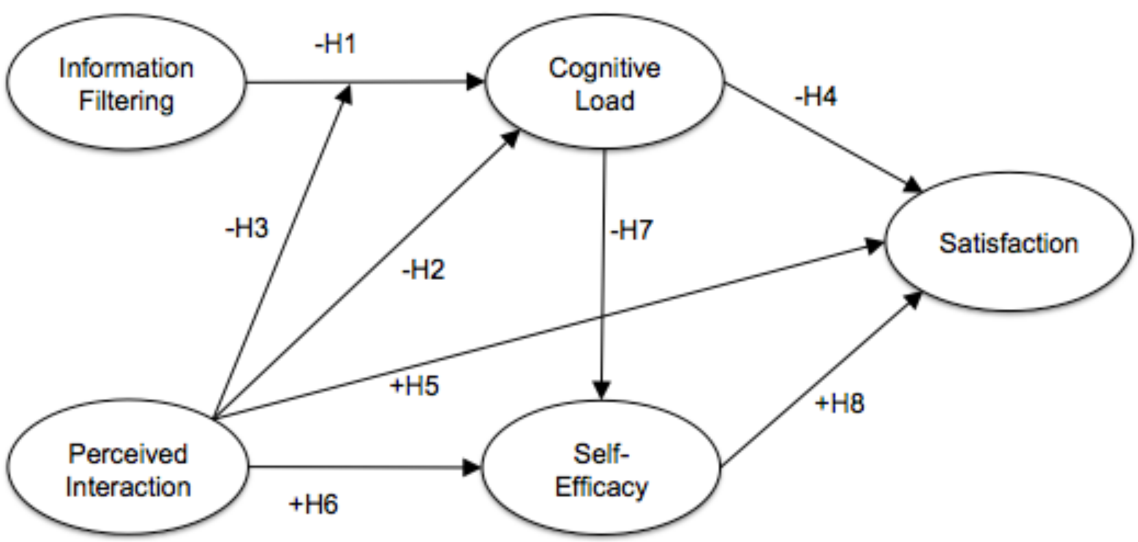


Figure 7: Research Model – Study Three

Hypotheses Summary

Table 1 summarizes the hypotheses used in the empirical evaluations in the three studies.

Table 1: Summary of Hypotheses

Study #	Hypotheses
1	1. Cognitive fit -> (+) coverage of domain ontologies created 2. Cognitive fit -> (+) relevancy of domain ontologies created 3. Cognitive fit -> (+) quality of domain ontologies 4. Cognitive fit -> (-) time to create domain ontologies 5. Cognitive fit -> (-) mental difficulty to create domain ontologies
2	1. Information filtering -> (-) cognitive load 2. Task complexity + Information filtering -> (+) cognitive load 3. Cognitive fit -> (-) cognitive load 4. Cognitive load -> (-) task performance.
3	1. Information filtering -> (-) cognitive load 2. Interaction -> (-) cognitive load 3. Information filtering + interaction -> (-) cognitive load 4. Cognitive load -> (-) satisfaction 5. Interaction -> (+) satisfaction 6. Interaction -> (+) self-efficacy

	7. Cognitive load -> (-) satisfaction 8. Self-efficacy -> (+) cognitive load

Research approach

The three studies presented here follow the design science research approach (Hevner et al. 2004) to build and evaluate the effectiveness of IT artifacts. Design research aims to understand, explain, and improve behavioral aspects of information systems with the analysis of the use and performance of designed artifacts. Our studies follow the seven guidelines articulated by Hevner et al. (2004). The problems that these studies attempt to solve are drawn from the real-world use of ontologies and are of interest to both IS research and practice. IT artifacts are created and rigorously evaluated based on relevant theories.

Experimental design

All of the three studies in the dissertation include theoretically grounded empirical evaluations of the IT artifacts developed. Study 1 involves an experiment with 1 x 2 factorial design with and without ontological knowledge. Study 2 involves an experiment in which a 2 x 2 factorial design in which information filtering and cognitive fit are used as treatments. Study 3 uses a 2 x 2 factorial design in which information filtering enabled by ontology and interaction are used as treatments. Pre-tests and pilot tests for these studies have been conducted to evaluate the research instruments, statistically validate the measures, and check manipulations (Straub 1989).

Analysis method

ANOVA and Partial Least Squares (PLS) analysis are used as the primary analysis tools in the three studies. One-way ANOVA is used in Study 1 to test for differences among two independent groups. It is used to assess the effect of ontological knowledge on domain ontology creation. PLS that is used in Study 2 and Study 3 is an advanced statistical method that allows optimal empirical assessment of a structural model together with its measurement model (Wold 1982). PLS analysis is considered appropriate because it places minimal demands on sample size and distributional assumptions (Chin 1998). PLS analysis is also appropriate for testing theoretical models in the early stages of development (Fornell 1982).

Conclusion

Although ontology has been studied from different perspectives in multiple academic disciplines, there has been no comprehensive attempt to study its creation, refinement, use and evaluation. This dissertation advances ontology research by developing methodologies for creation and pruning, creating prototypes and evaluating them using theoretically grounded empirical studies.

Each of the three studies has its own contribution. The first study provides a novel six-step methodology for ontology creation. The methodology is implemented in a prototype and is evaluated empirically. This study helps ontology engineers develop quality domain ontologies with use of the World Wide Web. The second study develops a methodology (implemented in a system) for ontology pruning and empirically evaluates its effectiveness. The third study provides an ontological approach to improving the retrieval and reuse of use cases. An interactive approach using ontology allows users to retrieve use cases accurately, thereby enhancing the reuse of use cases in large and complex system development projects.

Some of contributions are common to all the three studies. *First*, theoretically grounded studies are conducted for rigorous evaluation of the artifacts created in each study. This study draws on the following theories from cognitive science: cognitive fit, cognitive load, and interaction. *Second*, our research synergistically uses multiple research methods. Each of the three studies includes the design and development of IT artifacts in a form of prototype. These artifacts are evaluated using a laboratory experimentation, which provides a controlled environment to test theoretically grounded hypotheses. T*hird*, these studies are inter-disciplinary and draw from varied fields such as information systems, computer science, and biology. The IS perspective helps identify significant problems of interest to both research and practice that may be addressed using ontologies. Methods drawn from computer science are used to develop prototypes that address the challenges in using ontologies in the fields of biology and software development.

The remainder of this dissertation is organized into three chapters that present details of each of the three studies.

References

Burton-Jones, A., Storey, V.C., Sugumaran, V., and Ahluwalia, P. "A semiotic metrics suite for assessing the quality of ontologies," *Data & Knowledge Engineering* (55:1) 2005, pp 84-102.

Chin, W.W. "The partial least squares approach to structural equation modeling," in: *Modern Methods for Business Research,* G.A.Marcoulides (ed.), Lawrence Erlbaum Associates, Mahwah, 1998, pp. 295-336.

Conesa, J., and Olivé, A. "A General Method for Pruning OWL Ontologies," On the Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE (2), Agia Napa, Cyprus, 2004, pp. 981-998.

Conesa, J., and Olivé, A. "A Method for Pruning Ontologies in the Development of Conceptual Schemas of Information Systems," *Journal of Data Semantics*) 2006.

Corcho, O., Fernández-López, M., and Gómez-Pérez, A. "Methodologies, tools and languages for building ontologies. Where is their meeting point?," *Data & Knowledge Engineering* (46:1) 2003, pp 41-64.

Fensel, D., Van Harmelen, F., Klein, M., Akkermans, H., Broekstra, J., Fluff, C., Van Der Meer, J., Schnurr, H.P., Studer, R., and Hughes, J. "On-To-Knowledge: Ontology-based Tools for Knowledge Management," eBusiness and eWork 2000 Conf.(EMMSEC 2000), 2000.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. "METHONTOLOGY: From Ontological Art Towards Ontological Engineering," in: *Workshop on Ontological Engineering*, 1997

Fornell, C. *A Second Generation of Multivariate Analysis* Praeger New york, 1982.

Gangemi, A., Catenacci, C., Ciaramita, M., and Lehmann, J. "A theoretical framework for ontology evaluation and validation," Semantic Web Applications and Perspectives (SWAP) 2nd Italian Semantic Web Workshop, Trento, Italy, 2005.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. "Sweetening Ontologies with DOLCE," Proceedings of EKAW, 2002, pp. 166-181.

Gruber, T.R. "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition* (5:2) 1993, pp 199-220.

Gruninger, M., and Fox, M.S. "Methodology for the Design and Evaluation of Ontologies," Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI, 1995.

Guarino, N. "Towards a Formal Evaluation of Ontology Quality," *IEEE Intelligent Systems* (19:4) 2004.

Hevner, A.R., March, S.T., Park, J., and Ram, S. "Design Science in Information Systems Research," *MIS Quarterly* (28:1) 2004, pp 75-105.

Jacobson, I. *Object-oriented software engineering: a use case driven approach* Addison-Wesley, 1992.

Lee, J.-B., Kim, J.-j., and Park, J.C. "Automatic extension of Gene Ontology with flexible identification of candidate terms," *Bioinformatics* (22:6) 2006, pp 665-670.

Lenat, D.B. "CYC: a large-scale investment in knowledge infrastructure," *Communications of the ACM* (38:11) 1995, pp 33-38.

Maedche, A., and Staab, S. "Discovering conceptual relations from text," Proceedings of ECAI-2000, 2000, pp. 321-324.
Maedche, A., and Staab, S. "Ontology Learning for the Semantic Web," *IEEE Inteligent Systems* (16:1) 2001, pp 72 - 79.
Maedche, A., and Volz, R. "The ontology extraction and maintenance framework text-to-onto," Proceedings of the ICDM '01 Workshop on Integrating Data Mining and Knowledge Management, California, USA, 2001.
Noy, N.F., and McGuinness, D.L. "Ontology Development 101: A Guide to Creating Your First Ontology," KSL-01-05, Stanford Knowledge Systems Laboratory, Stanford.
Peterson, B.J., Andersen, W.A., and Engel, J. "Knowledge Bus: Generating Application-focused Databases from Large Ontologies," in: *KRDB'98*, 1998, pp. 2.1-2.10.
Pinto, H.S., and Martins, J.P. "Ontologies: How can They be Built?," *Knowledge and Information Systems* (6:4) 2004, pp 441-464.
Schreiber, G., Wielinga, B., and Jansweijer, W. "The KACTUS view on the 'O' Word," in: *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, 1995, pp. 159-168
Sengupta, K., and Te'eni, D. "Cognitive Feedback in GDSS: Improving Control and Convergence," *MIS Quarterly* (17:1) 1993, pp 87-113.
Straub, D.W. "Validating Instruments in MIS Research," *MIS Quarterly* (13:2) 1989, pp 147-169.
Studer, R., Benjamins, V.R., and Fensel, D. "Knowledge Engineering: Principles and Methods," *IEEE Transactions on Data and Knowledge Engineering* (25:1-2) 1998, pp 161-197.
Sutton, D.C. "Linguistic Problems with Requirements and Knowledge Elicitation," *Requirements Engineering* (5:2) 2000, pp 114-124.
Swartout, W.R., Patil, R., Knight, K., and Russ, T. "Toward Distributed use of Large-Scale Ontologies," in: *Proc. Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop, Canada*, 1996.
Sweller, J. "Cognitive load during problem solving: Effects on learning," *Cognitive Science* (12) 1988, pp 257-285.
Sweller, J., and Chandler, P. "Why Some Material Is Difficult to Learn," *Cognition and Instruction* (12:3) 1994, pp 185-233.
Tijerino, Y., Embley, D., Lonsdale, D., Ding, Y., and Nagy, G. "Towards Ontology Generation from Tables," *World Wide Web: Internet and Web Information Systems* (8) 2005, pp 261-285.
Tijerino, Y.A., Lonsdale, D.W., Embley, D.W., and Nagy, G. "Ontology Generation from Tables," in: *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE'03)*, 2003.
Uschold, M., Artificial Intelligence Applications, I., and University of, E. *Building Ontologies: Towards a Unified Methodology* Artificial Intelligence Applications Institute, University of Edinburgh, 1996.
Vessey, I. "Cognitive fit: A theory-based analysis of the graphs versus tables literature," *Decision Sciences* (22:2) 1991, pp 219-240.
Volz, R., Studer, R., Maedche, A., and Lauser, B. "Pruning-based Identification of Domain Ontologies," *Journal of Universal Computer Science* (9:6) 2003, pp 520-529.

Wold, H. "Soft modeling: The basic design and some extensions," in: *Systems under indirect observations: Part 2,* K.G. Joreskog and H. Wold (eds.), North-Holland, Amsterdam, 1982, pp. 1-54.

Chapter 2

Construction of Domain Ontologies: Sourcing the World Wide Web

Abstract

As the World Wide Web evolves into the Semantic Web, domain ontologies, which represent the concepts of an application domain and their associated relationships, have become increasingly important as surrogates for capturing and representing the semantics of real world applications. Much ontology development, however, remains manual and is both difficult and time-consuming. This research presents a methodology for semi-automatically generating domain ontologies from extracted information on the World Wide Web. The methodology is implemented in a prototype that integrates existing ontology and web organization tools. The prototype is used to develop ontologies for different application domains, and an empirical analysis is carried out to demonstrate the feasibility of the research.

Keywords
Ontology, ontology creation methodology, ontology evaluation, World Wide Web

1. Introduction

The World Wide Web is a massively distributed reservoir of information, but the information does not have well-defined, machine-understandable meaning attached to it, prohibiting automated manipulation and reasoning about such information (Ram et al., 2007). The next generation of the World Wide Web, the *Semantic Web*, is intended to enable more intelligent use of data and information for effective electronic interoperability and collaboration (Horrocks, 2008). A successful Semantic Web, however, depends upon the ability to manage, integrate, and analyze data and is driven by the role of semantics for automated approaches to exploiting Web resources (Berners-Lee et al., 2001). Ontologies, which are at the heart of the Semantic Web, define the concepts and relationships that make global interoperability possible, facilitate sharing and integration (Horrocks, 2008) and serve as surrogates for semantics. Ontologies are also useful for digital libraries and personalized information management (Katifori et al., 2007). Although their need is well-documented, ontology development is often performed manually and is challenging and time-consuming (Ding et al., 2002; Farquhar et al., 1997). One of the major reasons for this difficulty is finding relevant knowledge sources to use to create ontologies.

The World Wide Web is a great resource of information for almost all imaginable domains. If this information could be properly extracted and organized, it should be possible to effectively use it to create domain ontologies, especially if a process to do so could be automated to some extent (Sánchez et al., 2008). The objectives of this research, therefore, are to:

- develop a methodology for semi-automatically generating domain ontologies by extracting and organizing terms and relationships among those terms using the World Wide Web as a source;
- establish the feasibility of the ontology creation methodology by creating a prototype; and
- assess the performance of the methodology through an empirical analysis.

The contribution of the research is to develop a way to semi-automatically create domain ontologies by using the World Wide Web as a source and integrating web tools. Libraries could be used for the Semantic Web and other applications (e.g., heterogeneous databases, conceptual modeling, and web queries (Horrocks, 2008; Ram et al., 2007).

The next section examines related research on domain ontologies and its role in the Semantic Web. A six-step ontology creation methodology is presented in Section 3. Section 4 details the implementation of the methodology in a prototype, WebtoOnto. Section 5 evaluates the methodology using an empirical study. A summary and concluding remarks are presented in Section 6.

2. Related Research

2.1. Ontologies

An ontology is a way of describing one's world and can be used as a surrogate for semantics (Dahlgren, 1995). An ontology represents a set of concepts and the relationships among them for a specific domain. Ontologies have been developed in both Artificial Intelligence and knowledge management research to facilitate knowledge use and reuse with the main idea being to develop an understandable, complete, and sharable system of categories, labels, and relationships that represent the real world in an objective manner (Cristani et al., 2005; Horrocks, 2008). They are useful because they formalize a shared view of a domain. An example of an ontology for carpel tunnel syndrome (resulting from repetitive stress) created by our proposed methodology is shown in Figure 1.

There are a number of challenges to developing ontologies. Ontologies are specific to each domain and are time-consuming to create (Herman, 2007; Maedche et al., 2000a). Large-scale ontologies such as Cyc require a collaborative, community effort from knowledgeable people. Applications can be developed with small, domain specific ontologies (Herman, 2007), the creation of which is the focus of this research.

Organizations may use existing documents for domain ontology creation (Kietz et al., 2000; Maedche et al., 2000b; Sugiura et al., 2003). However, when they start a new business or expand an existing one, they may not have legacy resources upon which to draw. For example, when an organization develops a natural resource protection ontology to improve knowledge management and information sharing, they might have difficulty finding relevant knowledge sources for a specific species (Michener et al., 2007; Xing et al., 2009).

Domain ontologies specify concepts, relationship among concepts, and inference rules for a single application domain (e.g., airline reservations, art galleries, furniture, fishing, gourmet food) or task. They are not applicable across different domains; rather they capture agreed upon concepts, are applied to a specified context (Spyns et al., 2002), and are often created manually and collaboratively by domain experts (Noy et al., 2001).

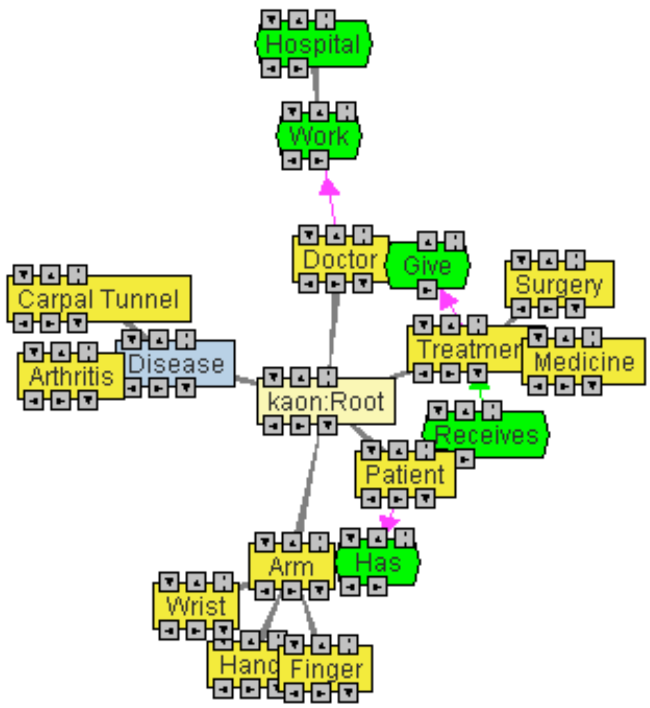


Figure 1: Carpel Tunnel Syndrome Ontology

2.2. Ontology Creation Methodologies

Ontology creation requires heuristics and expertise, rather than an engineering approach (Pinto et al., 2004). Prior research has concentrated on related tasks, such as ontology learning, ontology evaluation, evolution, and merging (Buitelaar et al., 2008; Corcho et al., 2003; Omelayenko, 2001). For example, ontology learning seeks to discover ontological knowledge from various forms of data automatically or semi-automatically using methods and tools such as

UIMA, GATE, OpenCalais, and WikiOnto (De Silva et al., 2009; Lau et al.; Zhou, 2007). Although prior research assumes that relevant information for ontology creation can easily be found, ontology developers may have difficulty doing so, especially for some specialized domains, which is one reason why this research focuses on the World Wide Web as a source.

Several ontology creation methodologies have been proposed. Zhou (2007) presents a framework for ontology learning, consisting of information extraction, ontology discovery, and ontology organization. Cristani and Cuel (2005) classify ontology creation methodologies (such as DOLCE (Gangemi et al., 2002), OTK (Fensel et al., 2000), TOVE (Gruninger et al., 1995), METHONTOLOGY (Fernández-López et al., 1997), Enterprise (Uschold et al., 1996)) as top-down and bottom-up. Top-down methods start with an abstract view of a domain and expand it with detailed specifications (e.g. KACTUS (Schreiber et al., 1995), DOLCE (Gangemi et al., 2002)). Bottom-up methodologies start from the specification of a certain task and obtain generalizations (e.g. TOVE (Gruninger et al., 1995), OTK (Fensel et al., 2000)). A middle-out method starts from the key concepts and moves to generalization and specialization (e.g. Enterprise (Uschold et al., 1996), METHONTOLOGY (Fernández-López et al., 1997)).

Ontology development may rely on a stage-based model (e.g. TOVE) and an evolving prototype (e.g. METHONTOLOGY). When the requirements and purposes of the ontology are specific and clear, the stage-based model is more appropriate than an evolving prototype, which is most useful when the environment is difficult to understand.

3. Ontology Creation Methodology

This section presents a six-step methodology for semi-automated ontology creation using terms from the World Wide Web. The methodology is heuristic in nature and takes advantage of existing tools. The methodology is based on a framework for ontology learning proposed by

Zhou (2007) and METHONTOLOGY (Fernández-López et al., 1997), which provides high-level steps for ontology creation as an existing partial solution. Our research expands and augments prior work and integrates tools. Figure 2 provides an overview of our methodology.

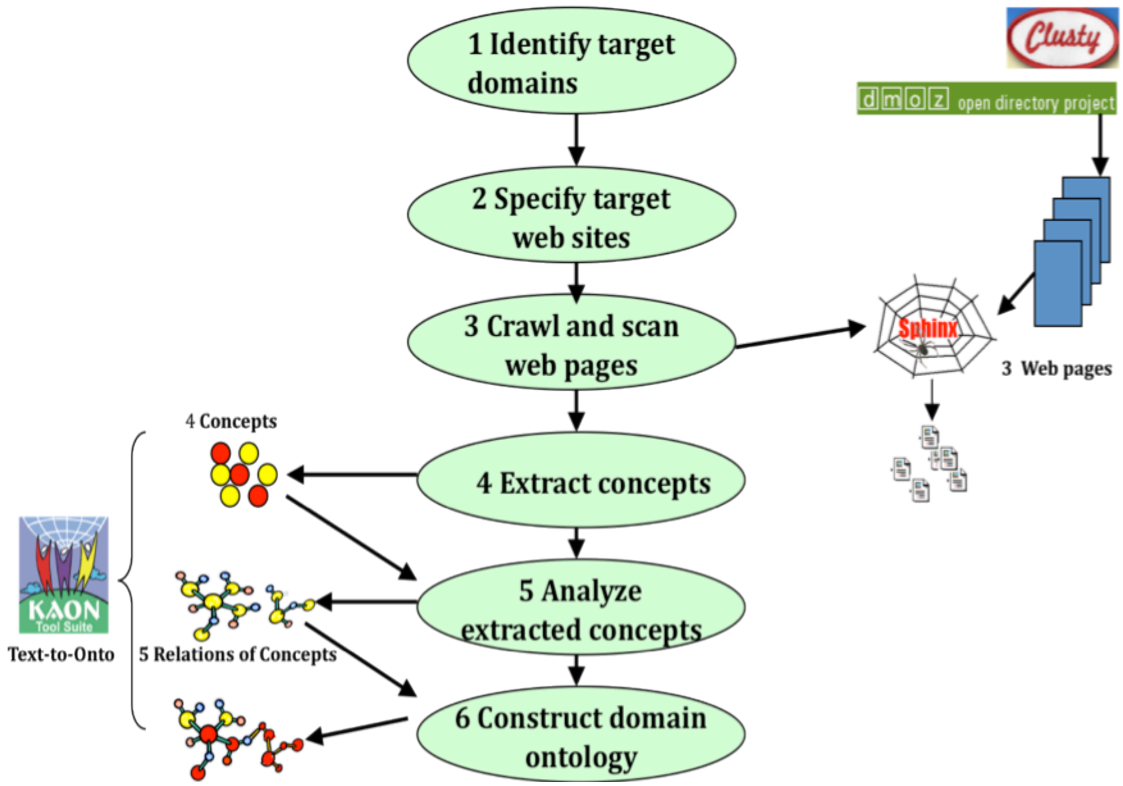


Figure 2: Methodology for Domain Ontology Creation

Step 1: Identify the scope of domains (Specification and initial conceptualization)

Application domains for which ontologies are needed may be of various sizes. Therefore, the first step in domain ontology creation is to identify the scope of the application domain (e.g., sports versus hockey versus Stanley Cup playoffs or gourmet dining versus gourmet food versus wine versus fine wines. This step is driven by the reason for the ontology development, intended uses, and potential users. The scope requires the identification of the categories of the domains in which the users are interested.

Figure 3: Results from DMOZ and Clusty

Several attempts have been made to categorize the vast and diverse web pages on the World Wide Web into domains (Boley et al., 1999a; Boley et al., 1999b; Chakrabarti et al., 1999), motivated by the fact that search engines are often unable to provide content-dependent, useful results (Fagni et al., 2006). Two major business approaches to web site classification are the interactive tools, DMOZ (www.dmoz.com) and Clusty (www.clusty.com). DMOZ (Directory Mozilla) is an Open Directory Project that attempts to create the most comprehensive human edited directory of the web. DMOZ provides meta-level categorization, expressed in XML. This directory is created and managed by net-citizens' voluntary participation. Each citizen organizes a small portion of the web by removing useless content. When users initiate a search query, DMOZ provides a list of categorized web sites. For example, in Figure 3, when wine is provided as a keyword, DMOZ displays categories related to wine and corresponding web sites.

Clusty hierarchically clusters terms used on web sites by topics and URLs. It uses a clustering engine to organize results from search engines, such as MSN and Lycos, into folders, grouping similar web sites together. Its clustering algorithm puts search results together based on textual and linguistic similarity. Clusty allows users to obtain a quick overview of the domains associated with a given query.

To select the categories of the target domains, users identify and choose relevant categories provided by either DMOZ or Clusty. It also involves specifying key concepts for constructing a target domain. For a small example (for illustration purposes) of a wine ontology, the first two categories (Recreation: Food: Drink: Wine and Shopping: Food: Beverage: Wine) of DMOZ and the 'white wine' section of Clusty can be identified as relevant and selected as illustrated in Figure 3. The key concepts are identified by searching through categories and related topics from DMOZ and Clusty. For example, 'food' and 'drink,' as shown in Figure 3, can be selected as important terms. The keywords selected from DMOZ and Clusty are the initial key concepts.

Step 2: Specify target web sites

The user specifies target web sites within a domain or category selected from Step 1.

Both DMOZ and Clusty identify a set of web sites appropriate for a given domain. DMOZ shows the conceptual hierarchical structure of terms and related web sites. Clusty provides web pages based on two or three-level clustered terms. Using DMOZ and Clusty, the user selects target web sites related to a given domain. Some categories may have many relevant web sites (e.g., over one hundred). Thus, it is not practical to specify all websites for each category, so a user needs to browse the web sites before selecting relevant ones to ensure a high level of quality and relevancy of the chosen websites.

The purpose of this step is to provide the basic resources for the next four steps. When a user selects web sites relevant to a domain, the user will have a better chance of collecting relevant terms and creating domain ontologies with high quality. This is to deal with the well-known context problem (Gu et al., 2004). If there were no interaction with the user for the selection of the websites, the created ontologies would not be context-dependent, which is a key required characteristic of a domain ontology. However, too much interaction with the users would increase the required time and effort. Since categorized websites can assist users, users can browse and select relevant ones from DMOZ and Clusty, thus, providing well-organized websites for a selected domain.

Step 3: Crawl and Scan Web Pages

Related web pages need to be selected based on the results from Step 2. WebSphinx (http://www.cs.cmu.edu/~rcm/websphinx/) crawls and scans web pages from the selected websites at DMOZ or Clusty. The user can specify the scope and depth of crawling. The scope refers to the range of data collection. The user can set the scope as a sub directory of the website, server, or web. The sub-directory option restricts WebSphinx to crawling only the lower levels of the selected web address. Based on the scope of the server, crawling is constrained within the server.

When the user selects the web as its scope, WebSphinx crawls documents outside of the server. The depth of crawling refers to the number of hops. For example, when the number of hops is three, WebSphinx will crawl all of the three lower levels. The number of hops, thus, limits the depth, as well as the scope, of crawling.

To obtain web pages related to the selected web sites, the user sets the scope as the server of the target web pages. This setting allows WebSphinx to collect web pages within a specified website without scrawling beyond that website as shown in Figure 4. WebSphinx stores web pages as html or txt file format. The scope of the selection is important in controlling the content and the amount of web documents WebSphinx collects.

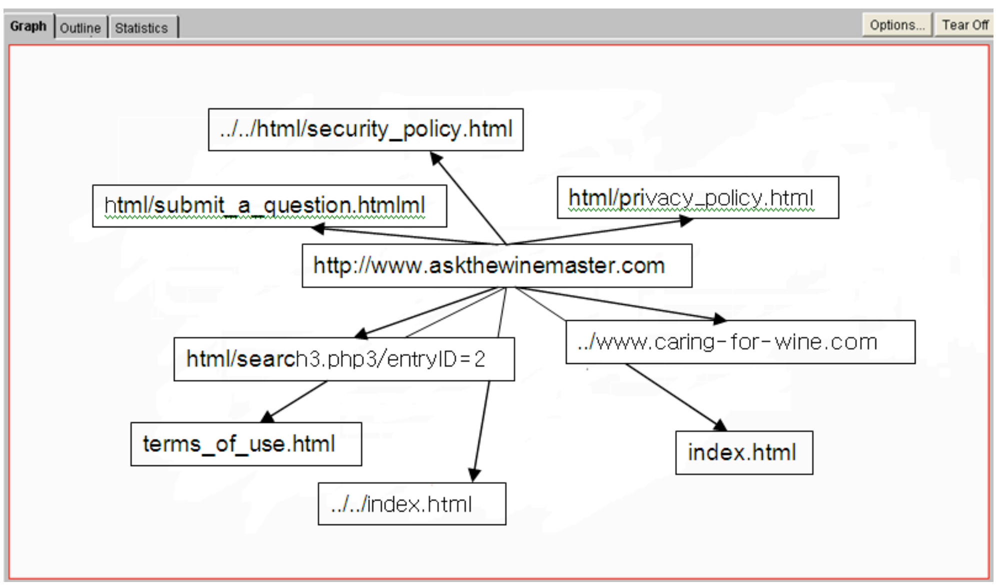


Figure 4: Crawling and Scanning by WebSphinx

Step 4: Extract Concepts (Conceptualization)

Candidate terms for ontologies are selected at this step by user. The user browses the results of Step 3 and selects candidate terms related to the domain based upon the importance and the relevance of the term to the domain ontology. Text-to-Onto (Maedche et al., 2001) assists the user in the extraction of the concepts by providing relevant information.

The text or html files stored by WebSphinx provide input to Text-to-Onto which provides support for ontology creation from texts. Text-to-Onto is a module of KAON (Karlsruhe Ontology and Semantic web infrastructure), an open-source ontology management infrastructure targeted for business applications (Maedche et al., 2001). Text-to-Onto is built on three text mining algorithms: a term extraction algorithm, a concept association extraction algorithm, and an ontology pruning algorithm. It also supports a graphical interface and stores a generated ontology such as XML (RDF schema format). Text-to-Onto constructs an ontology from domain-specific text using machine learning techniques and algorithms. It extracts terms and provides users with information such as frequency, Term Frequency Inverse Document Frequency (TFIDF), Entropy, and C-value.

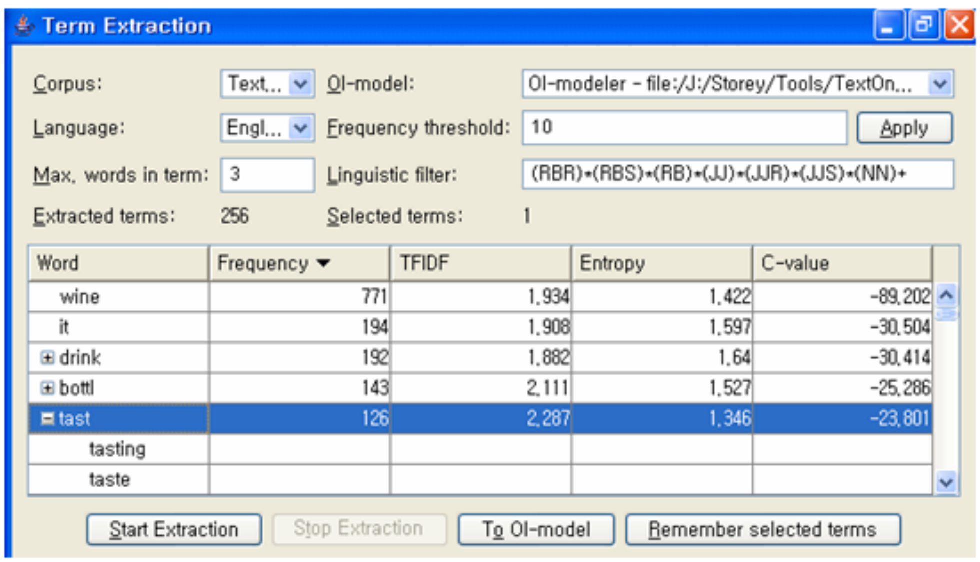


Figure 5: Term Extraction using Text-to-Onto

With this information, relevant ontology terms, as shown in Figure 5, can be selected by the user. TFIDF shows how important a selected term is within a document (Salton et al., 1988). Entropy indicates the rate of disorder of words in a document. The C-value (Collocation-value) improves the extraction of nested multi-word terms and collocations (Frantzi et al., 1999) in a domain-independent manner by combining linguistic and statistical information retrieval techniques. The higher the C-value, the greater the likelihood of a candidate term being a valid term. For example, 'shark species' has a higher C-value than 'seafood dealer' in shark-related documents because 'shark species' is more important than 'seafood dealer'. Relevant terms should be selected by a human user who can understand the domain and the context. Selected concepts are then used in the domain ontology construction.

Step 5: Analyze and Cluster Extracted Features (Conceptualization)

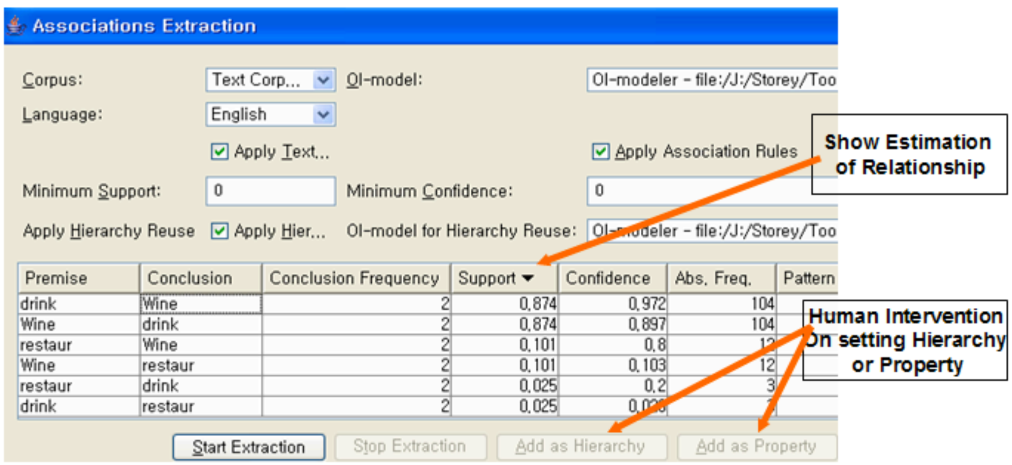


Figure 6: Association Extractions by Text-to-Onto

The purpose of this step is to analyze terms and identify relationships among selected terms. Text-to-Onto provides information on the relationships between two terms using its association rules extraction and linguistic patterns. Figure 6 shows that drink and wine have a strong relationship, whereas the relationship between restaurant and wine is weak. Based on these estimations, a user can add terms as hierarchical relationships or properties to begin the ontology construction. Although Text-to-Onto can help users identify and analyze the relationships between terms, users may need to modify the relationships for a specific context. For example, users might want to manually establish a property relationship between restaurant and drink, even though the computed values of support and confidence are low. During this step, the relationships among selected concepts are established to build domain ontologies. This corresponds to the conceptualization process in METHONTOLOGY.

Step 6: Construct Domain Ontology (Formalization and Implementation)

During this step, a domain ontology is actually constructed using terms selected from step 4 and information about the terms provided from step 5. Text-to-Onto supports a graphic interface and a feature to store constructed ontologies in RDF. A portion of a constructed wine ontology is shown in Figure 7. This ontology captures high level concepts related to wine and meat. If the ontology creator is an expert in wine, the creator can expand the branches of red wine and white wine. Price is included as a property of drink, although it would be better for it to be a property of food or drink. Restaurant is also added because it serves both wine and meat. It is linked to ‘Food or Drink’ by the property ‘serve’.

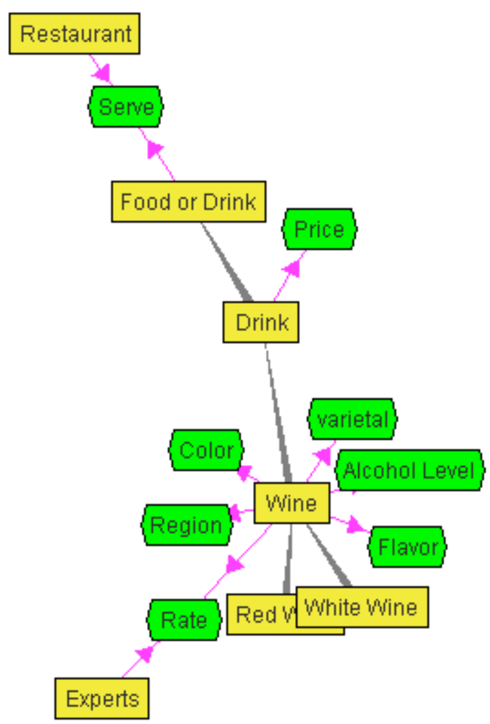


Figure 7: A Part of Wine Ontology as represented by Text-to-Onto

This step corresponds to formalization and implementation suggested by METHONTOLOGY. Users transform the conceptual model into a formal, computable model using the application that supports this step by converting the conceptual model into an XML format.

4. Implementation

The ontology creation methodology has been implemented in a prototype developed as a window program in Java, the architecture of which is shown in Figure 8. The prototype, called WebtoOnto, is comprised of three modules: Category Retrieval Module, Web Crawler Module and Ontology Creation Module. The purpose of the prototype system is to demonstrate that the methodology is feasible and use it as a test-bed for empirical assessment and future enhancement.

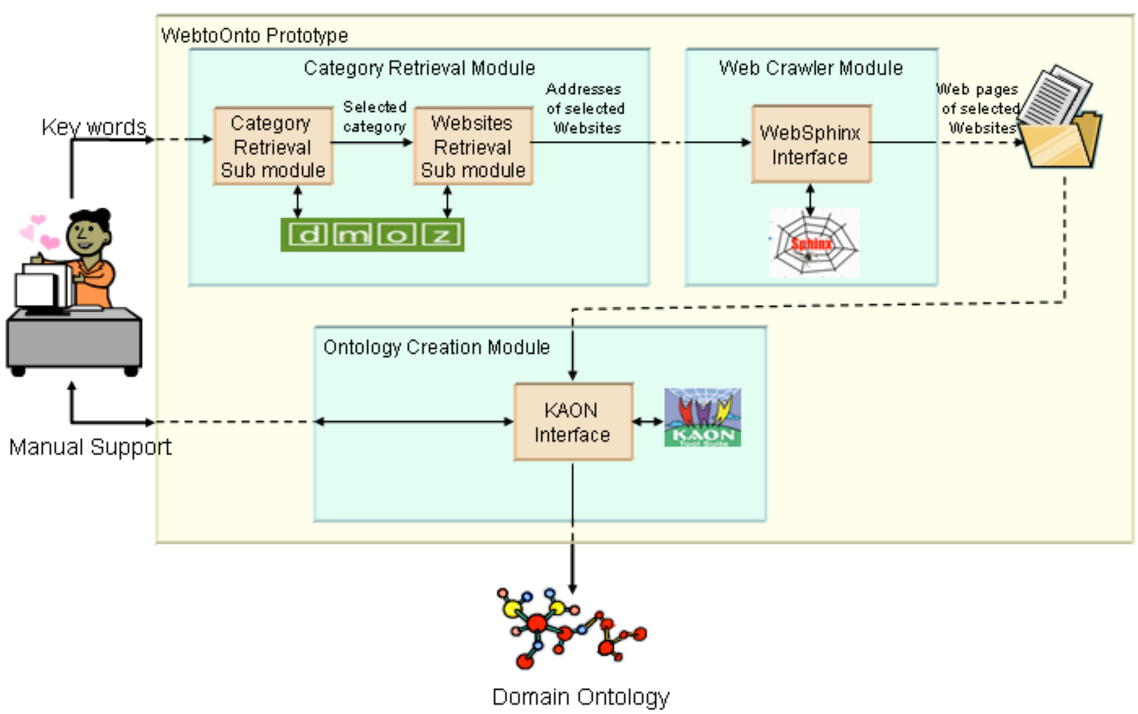
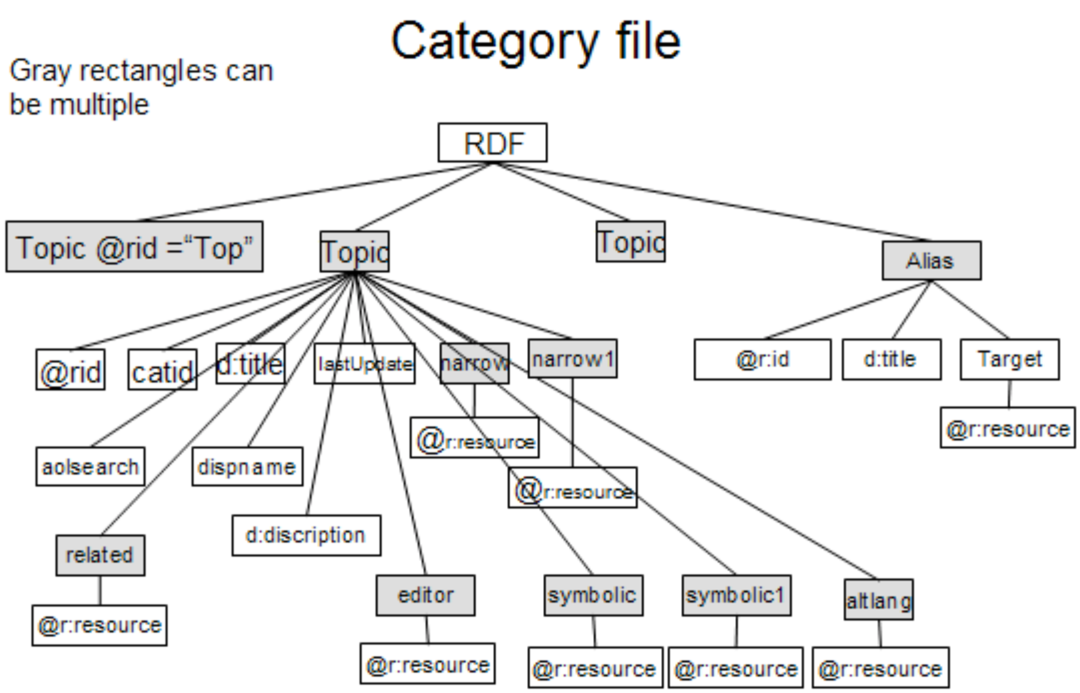


Figure 8: System Architecture of WebtoOnto

DMOZ is used in the category retrieval module because it integrates well. The category retrieval module has two sub-modules: a category retrieval sub-module and a website retrieval sub-module, both of which use RDF data files downloaded from the DMOZ website. One RDF file contains website category hierarchy information. The other holds web page link information within each category. Figure 9 shows the schema of these two files.

4.1 Category Retrieval Sub Module

The category retrieval sub-module receives keywords from the user and arranges the strings to be queried onto the RDF file. This module incorporates Step 1 of our proposed methodology. The user interacts with this module to select key terms that set the scope of the domain ontology. This sub-module submits the corresponding category list to the user who then selects the categories of interest based upon key terms.



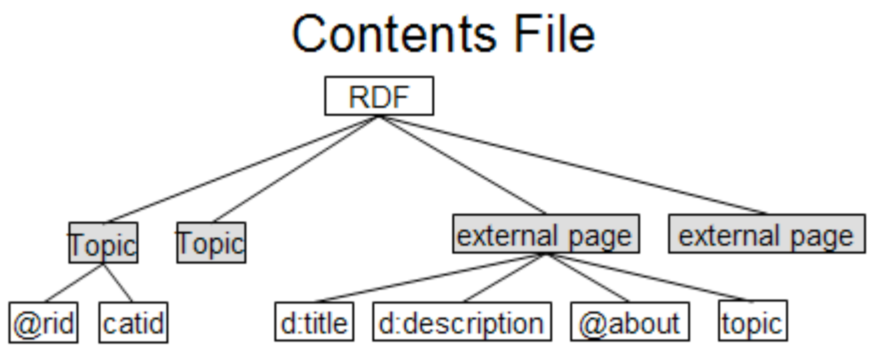


Figure 9: Schema of DMOZ Category and Contents File

Four flags are implemented to identify the topic from the keywords and to mark the acquisition of the flagged attributes. Four elements (topic, title, description and external) are used to set the flags. Using the Simple API for XML (SAX) parser, this sub module queries the web sites' descriptions provided by DMOZ and receives notification of the XML parsing result. The SAX parser, an event-driven parser, is used because it is faster and has more efficient memory use than the Document Object Model (DOM)-style parsers. The return is comprised of a URL, local name, q-name, and parsing exception. Each entry of a result obtained uses a vector to store the nodes of web pages because a topic can have multiple, related web pages.

4.2 Website Retrieval Sub module

This module receives a list of input categories selected by the user as shown in Figure 10. To provide further support, WebtoOnto allows the user to view the selected web pages. This enables the user to make better decisions on relevant websites by allowing the user to view multiple sites before making a selection, saving time and effort. WindowSwitcher is implemented to support this feature.

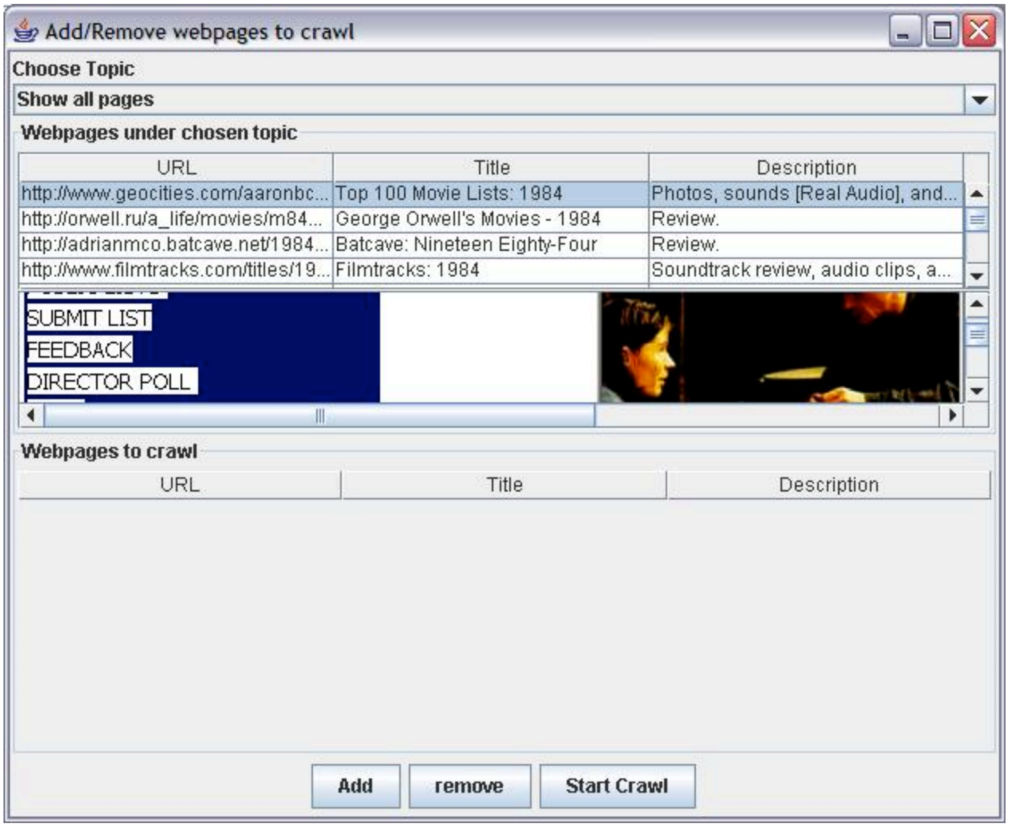


Figure 10: Website Retrieval Sub Module

4.3 Web crawler module

The website retrieval module sends the addresses of the selected websites to the web crawler module. Using the APIs provided by WebSphinx, this module retrieves the corresponding web pages of the selected sites and stores these files in either html or txt format on the local hard drive.

4.4 Ontology creation module

The ontology creation module, using Text-to-Onto, receives the stored files as input and handles the ontology creation processes from Step 4 to Step 6. Each step is partially automated. For example, the term extraction process is automated with the use of a POS (Part-Of-Speech) tagger (Banko et al., 2004). In addition, information such as TFIDF supports the analysis of the selected terms. Finally, the ontology construction is supported by a graphical interface and a RDF conversion feature.

The prototype is an integrated tool for organizations to develop domain ontologies from web documents. It is intended to minimize the effort required for ontology creation with the use of information found in web pages. This tool can help users quickly identify the relevant web pages of a target domain, process them using WebSphinx, and create domain ontologies using Text-to-Onto. It is possible that WebtoOnto could be integrated with existing ontology engineering tools. For example, the category module and website retrieval module of WebtoOnto might serve as a plug-in to Protégé.

5. Evaluation

Evaluation of our research is two-fold. First, the feasibility of the six-step methodology was tested by developing the WebtoOnto prototype, so it can be used by professionals. Second, to assess WebtoOnto's utility and performance in developing domain ontologies, an empirical analysis was carried out. Several hypotheses, based on Cognitive Fit Theory (Vessey et al., 1991), were developed and tested in an experimental setting. Even though the proposed methodology can be implemented, it is, of course, another matter to assess whether it is useful. Thus, to assess the usefulness of the methodology, a laboratory experiment was carried out in which ontologies created by two groups were compared.

5.1 Performance Test

5.1.1 Hypotheses

This study employs Cognitive Fit Theory (CFT) (Vessey et al., 1991) to assess the performance of ontology creation with/without information from WebtoOnto. CFT explains how information representation affects the decision processes and decision-making outcomes. CFT has been applied to various areas of information systems, including decision making in geographic information systems (Dennis et al., 1998), consumer learning and shopping behavior in e-commerce (Hong et al., 2004; Suh et al., 2005), and software engineering (Shaft et al., 2006).

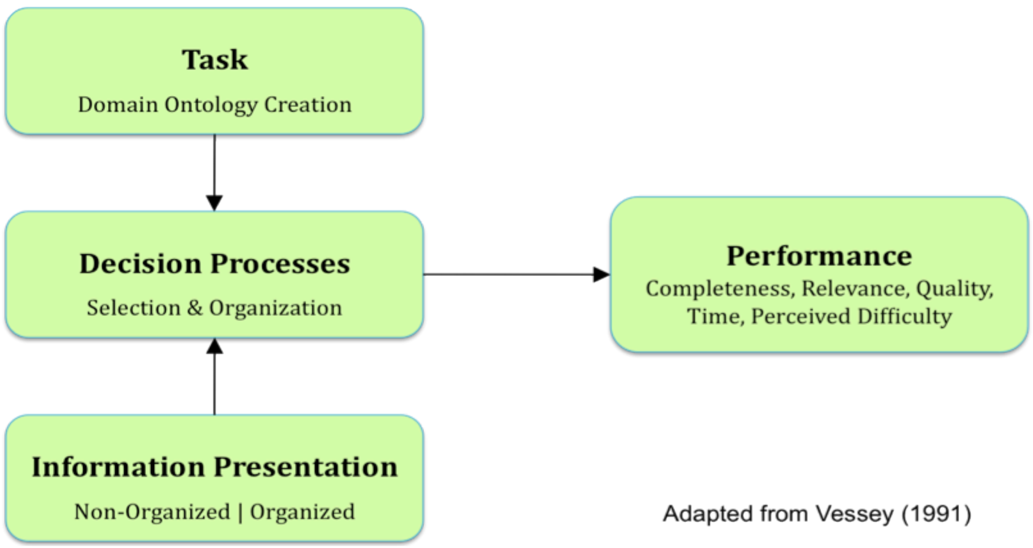


Figure 11: Cognitive Fit for Ontology Creation Task

According to CFT, decision makers develop a mental representation of the task and adopt decision processes based upon the task and presentation of task information as shown in Figure 11. Vessey (1991) argues that decision makers can deliver faster and more accurate solutions when the presented information matches the mental representation of the task. This is because the decision makers use the same mental representation and decision processes for both the representation and the task. Thus, the fit between information presentation, task, and decision processes may affect performance.

Cognitive Fit Theory can be applied to a multi-criteria task such as domain ontology creation which evaluates several alternatives based upon a set of criteria. For example, ontology creators must evaluate and select terms, and then organize the selected terms to represent a given domain. Domain ontology creation is neither a spatial, nor a symbolic task; instead, it is more cognitively intensive. Therefore, well-organized information that supports an ontology developer's mental representation should improve ontology creation. The World Wide Web contains much information that could help an ontology developer, but requires an ontology creator to search through a great deal of irrelevant information. The information provided by WebtoOnto, however, is well-organized, thereby supporting the mental representation needed for the task of ontology creation.

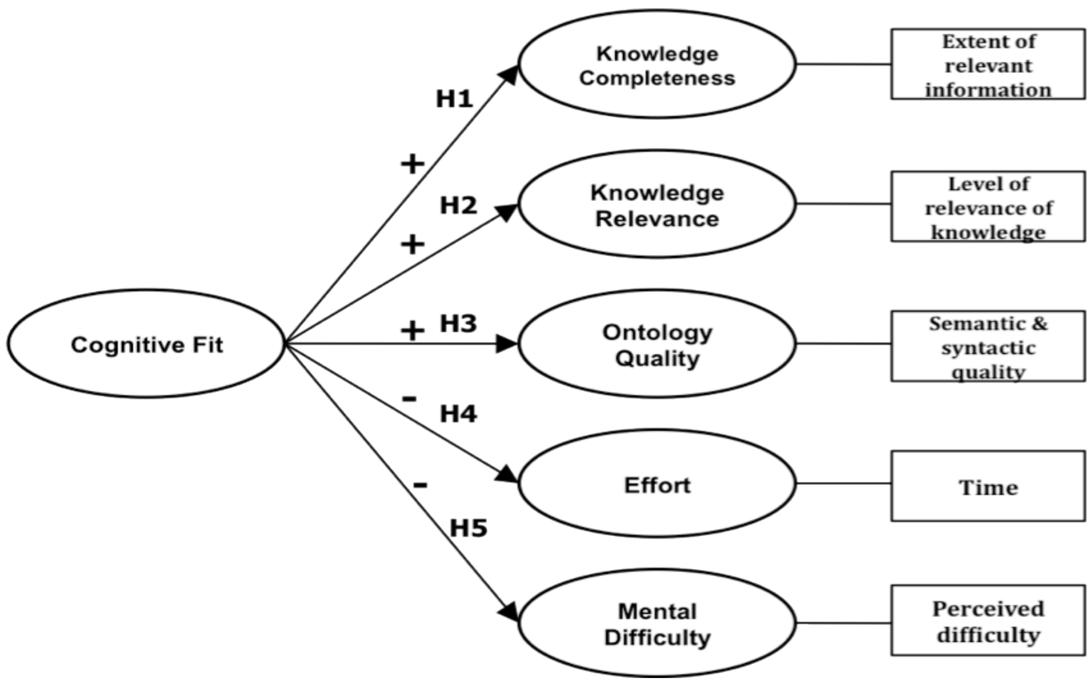


Figure 12: Research Model

Five hypotheses, shown in Figure 12 and Table 1, were proposed and tested. In Figure 12, constructs are shaped as ovals and elements as rectangles. From step 5, our approach is intended to compare two groups: 1) a control group with ill-represented information, and 2) a treatment group with well-represented information for domain ontology creation. The treatment group received a small number of selected terms with information for ontology developers to refer to when creating a domain ontology. Providing these terms should help ontology developers create ontologies with better quality and less time and effort than searching the World Wide Web.

The five dependent variables representing performance are knowledge completeness (KC), knowledge relevance (KR), ontology quality (OQ), time, and perceived difficulty (Burton-Jones et al., 2005; Dweck, 1986; Lindland et al., 1994; Paas, 1992; Steinberg, 1989). Knowledge completeness measures the extent of relevant information captured in a domain ontology. Knowledge relevance measures the level of relevance of the knowledge represented, and ontology quality measures the semantic and syntactic quality of a domain ontology. For example, correctness and meaningfulness of inheritance relationships and relevant properties are used for ontology quality measurement. Time and perceived difficulty are measured to represent resources used to develop domain ontologies.

Table 1: Five Hypotheses

#	**Hypotheses**
H1	Cognitive Fit is positively associated with Knowledge Completeness
H2	Cognitive Fit is positively associated with Knowledge Relevance
H3	Cognitive Fit is positively associated with Ontology Quality
H4	Cognitive Fit is negatively associated with Effort.
H5	Cognitive Fit is negatively associated with Mental Effort.

The five hypotheses, shown in Table 1, are based on Cognitive Fit Theory about the relationship between the ontology creation task and information provided by the prototype.

5.1.2 Design

A laboratory experiment was used to test hypotheses H1-H5. This methodology helped to control other factors that might impact a subject's ontology creation.

The experiment used a 1*2 between-group design as shown in Table 2. The control group received instructions that contained information on ontology creation and was asked to search the World Wide Web to find relevant concepts for the ontology. The treatment group received the same instructions on what constitutes a domain ontology and the subjects were asked to create one (See Appendix A). Rather than being asked to search the World Wide Web, the subjects in the treatment group received a document containing terms identified by the methodology. The participants were randomly assigned to two groups.

Table 2: 1*2 Between-Group Design

Information Presentation	
Control Group (Non-Organized)	Treatment Group (Organized)
Internet	A table-format data

A total of 60 students from information systems classes at a large U.S. university participated. All of the subjects were familiar with entity-relationship (ER) diagrams from their coursework. ER diagrams are assumed to be a reasonable precursor to understanding domain ontologies and regarded as a conceptual model shared by stakeholders (Motik et al., 2002). Thus, subjects with knowledge of conceptual modeling, such as ER diagrams, should be able to understand and learn how to create an ontology quickly. The control and the treatment groups were equally, and randomly, divided into 30 subjects.

Four of the five dependent variables (all except time) were evaluated using a seven-point scale. The domain ontology diagrams created by the subjects were evaluated by an ontology expert who assessed each domain ontology diagram based upon the dependent variables: KC, KR, and OQ (see Appendix B). The ontology expert holds a Ph.D. in computer science and has conducted research on ontology creation, ontology integration, and ontology engineering (pruning and refactoring of ontologies). He has applied ontologies to support conceptual modeling activities, e-learning (using upper-level ontologies to enhance the description of learning objects and creating ontologies to define and execute learning processes) and web searches (using very large ontologies to support the disambiguation and expansion of web queries). He teaches conceptual modeling and the semantic web, including evaluating hundreds of students' conceptual models (and ontologies). Thus, the expert was qualified to make a valid assessment.

The expert did not know from which group a domain ontology diagram came. The seven-point scale for these three variables ranged from "very low" (1) to "very high" (7). For time and perceived difficulty, the scores reported by subjects were used. Perceived difficulty was measured using two items that were anchored on a seven-point scale ranging from "very little" (1) to "very much" (7).

5.1.3 Procedure

Five students participated in a pilot study. Minor modifications were made to the materials and procedures based upon their feedback. The undergraduate students were given the materials after their class had studied entity-relationship diagrams. This was because studying and using entity-relationship diagrams gave the students experience modeling the real world and representing it in a manner that captures concepts and associations between them. The experimental task was given as an assignment to the students who received participation credit. The assignments were completed within one week.

5.1.4 Results

Table 3 and 4 show the results. One subject neglected to answer questions on perceived difficulty so the values of the subgroup mean were used (Tsikriktsis, 2005). The ANOVA results that tested the hypotheses are shown in Table 1. The first three hypotheses receive support from the data. The differences between the control and treatment groups were significant across three dependent variables: KC, KR, and OQ. The treatment group, using the data created by our approach, exceeded the control group in these three areas. For the Time variable, the control group spent more time creating a domain ontology than the treatment group. However, the difference in time was not statistically significant. As for Perceived Difficulty, the treatment group perceived more difficulty in creating a domain ontology than the control group, so H5 is not supported.

Table 3: Average Values of Two Groups

	Individuals	KC	KR	OQ	Time	PD
Control Group	30	2.43	4.23	3.53	23.83	4.22
Treatment Group	30	4.93	5.83	4.30	20.53	4.79

Table 4: ANOVA Results

Dependent Variables	df	Mean (Standard Dev.) Between Groups	F	Sig.
KC	58,1	4.93 (1.43) (Treatment) > 2.43 (1.10) (Control)	57.09	.00
KR	58,1	5.83 (0.98) (Treatment) > 4.23 (1.65) (Control)	20.71	.00
OQ	58,1	4.30 (1.36) (Treatment) > 3.53 (1.63) (Control)	3.88	.05
Time	58,1	23.83 (14.32) (Control) > 20.53 (11.40)(Treatment)	0.97	.24
PD	58,1	4.79 (1.65) (Treatment) > 4.22 (1.20) (Control)	2.39	.13

5.2 Discussion

The results of the lab experiment show that the treatment group, working with information from WebtoOnto, created ontologies with better quality than those created by the control group. The first three hypotheses support this finding. For the fourth hypothesis on the time variable, the control group perceived more difficulty than the treatment group. But this difference is not significant. A possible explanation for this result includes the simplicity of the given task, and that more data is needed to establish a statistical difference. However, with the quality being better, even for the same time, the overall approach is worthwhile.

Surprisingly, the fifth hypothesis on the perceived difficulty variable was not supported. A possible reason is that the treatment group was more engaged in organizing terms than the control group. Qualitative verbal data collected from the subjects support this speculation. Only 30% (9/30) of the control group subjects mentioned that term organization was a difficult task whereas 80% (24/30) of the treatment group subjects identified term organization as difficult. Term organization is a more cognitively complex task than web searching or term selection. In that sense, it is understandable that the treatment group perceived more difficulty than the control group. Subjects might have perceived the ontology creation as a relatively simple task to receive an extra credit. Therefore, they might have set a certain time limit for the task. Whereas the treatment group spent time on term organization, the control group focused on term identification, which was time-consuming, but less cognitively complex. This issue could be addressed by constraining the number of terms in the ontology. Another possibility is that the users found the interface of the prototype difficult to manage, which could be addressed with a more complete prototype instead of the proof-of-concept one used in the research (which still improves quality). Finally, the subjects might have been able to formulate associations between categories easier without the constraints of the user interface and, perhaps, similar to their training on entity-relationship diagrams.

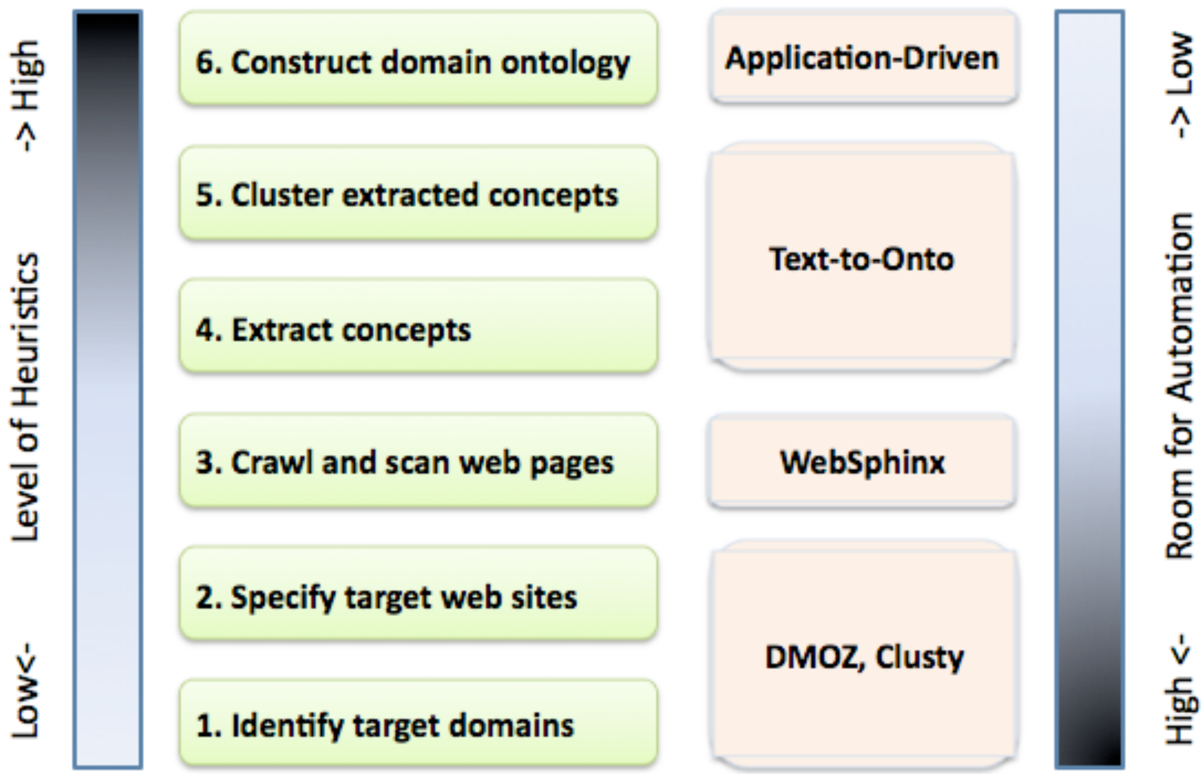


Figure 13: Analysis of Ontology Generation Methodology Steps

Although the overall goal of this research is to develop a methodology for automating domain ontology development, complete automation was not achievable (Zhou 2007). However, the tool improves KC, KR, and OQ, suggesting that it is most useful to ontology developers to support term selection and organization in the ontology creation process. With regard to term selection, high scores in KC and KR mean that the scope and relevancy of the terms provided by the tool are adequate to develop domain ontologies. As a result, ontology quality in terms of semantic and syntactic quality is high as engineers can better organize relevant terms.

The right rectangle in Figure 13 indicates how much further automation is possible with current tools. Step 1 and Step 2 require human interaction, although DMOZ and Clusty provide support for these steps. Thus, this work successfully integrates existing, separate tools to minimize interruption and improve the related processes.

5.3 Limitations

The empirical analysis has two main limitations. First, our experiment limits the source to the World Wide Web and a given document to control and treatment groups, respectively, in order to represent a real world context. An attempt was made to be as realistic as possible in creating an initial domain ontology and to control extraneous sources of variance by asking subjects to create an ontology for an unfamiliar domain. Second, the use of student subjects can limit the generalizability of the results. However, student subjects are commonly used in experiments that probe human decision-making (Harrison et al., 1993; Sitkin et al., 1995). Ontology creation requires a series of decision-making in terms of selection and organization. Moreover, when professionals need to develop an ontology for an unfamiliar domain, students and professionals are in a similar situation. In this sense, each subject could reasonably represent a professional, such as a software engineer or an ontology developer.

The first three propositions are supported by the data. The correlation analysis (Table 5) shows that KC, KR, and OQ are significantly correlated. The correlation between KC and OQ is less significant than the other two correlations: KC vs. KR and KR vs. OQ. When the knowledge completeness of the ontologies is high, this means that the created ontologies cover a wide range of selected domains. When ontologies consist of relevant terms, the semantic and syntactic quality of the ontologies is assessed to be high. Further research is needed to analyze these relationships. In addition, the respondents' prior knowledge about a given domain is difficult to control. To address this issue, an unfamiliar domain was chosen for which respondents were asked to construct ontologies. Of course, even the selected topic might be familiar to certain subjects on some level. Finally, terms with only TFIDF values were given to the treatment group. This was to avoid providing too much information for the treatment group to process during the experiment.

Table 5: Pearson Correlation Results

Constructs	KC	KR	OQ	Time	PD
KC	-	.648**	.138	.060	.186
KR		-	.455**	.075	.170
OQ			-	.091	.127
Time				-	.337**
PD					-

**: Correlation is significant at the 0.01 level.

There are several feasible explanations for the insignificant difference in time taken to create a domain ontology. The most likely is a technical explanation. The file size was close to three gigabytes and retrieving relevant information on the domain structure website from Dmoz's RDF files took more than 10 minutes. With more computing power, this problem could easily be addressed. Another technical issue was that some web pages could not be retrieved when they were developed in programming languages (e.g. ASP and PHP). When users are aware of this issue, they can avoid using these websites as the source of their domain ontologies. Finally, there might have been early "giving-up" by members of the control group due to cognitive difficulties.

5.4 Contribution

There are two main contributions of this research. For practitioners, the six-step methodology and WebtoOnto can help improve ontology creations. The methodology also provides guidance for using the World Wide Web as a source for creating domain ontologies. From a research prospective, this study uses Cognitive Fit Theory to evaluate how the format of information can affect task performance within the context of ontology development. It does so by highlighting how well-organized web sources improve the development of domain ontologies.

6. Conclusion

This research has presented a methodology and prototype implementation for semi-automated ontology generation. The methodology identifies relevant web pages for domain ontology creation, and for extracting terms and relationships from them. This methodology is intended to contribute to the interdisciplinary effort to the World Wide Web as it matures into the Semantic Web through the help of ontologies (Hendler et al., 2008). The prototype integrates a variety of tools to demonstrate how ontology creation can be semi-automated. An empirical assessment revealed support for the effectiveness of the methodology, which may help users create better quality domain ontologies by enabling them to select relevant terms quite easily and focus on organizing them.

Further work is needed to enhance the prototype and to create libraries of ontologies. For example, ontology sources such as DBpedia/Wikipedia and the Linked Data Web could be used. The ontology creation methodology could be integrated with web query tools to provide a more complete solution. Finally, the domain ontologies could be coupled with other repositories of knowledge and applied to various applications.

APPENDIX A

Experimental Instructions Given to Treatment Group

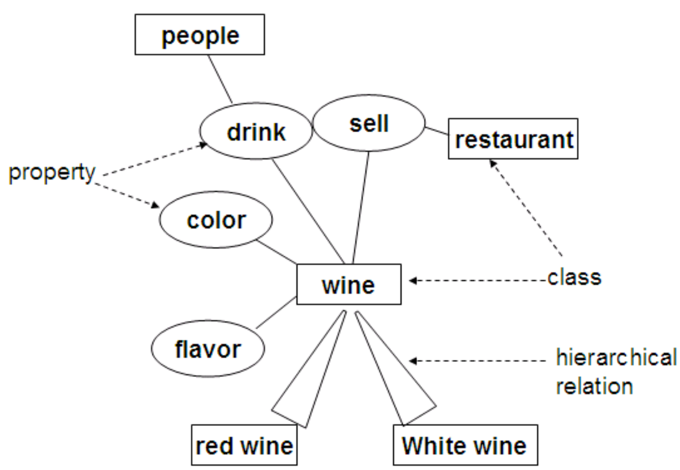
Please read the instructions and follow the tutorial presented below.

Instructions

You are going to develop an ontology about “**Shark and its Conservation**” which are **two main topics** for your ontology. For this exercise, 1) **use the attached document to select important terms** (you can come up with terms which are not included in the given document), then, 2) **create your mind map (also called an ontology) by following the tutorial below**.

Tutorial for Ontology Creation

1. Determine important **terms** (can be either classes or properties) in ontology
   - `Example terms in Wine ontology: wine, color, flavor, sell, restaurant`
2. Define the **classes (rectangle)** from identified terms and the class hierarchy
   - `Example of wine subclasses: white wine, red wine`
3. Define the **properties (oval)** of classes
   - `Example of wine properties: color, flavor`



4. Classes can be linked (**line**) by shared properties (example: sell)
5. The attached document contains terms extracted from related web pages. You are supposed to select terms from the document which is the main source of the terms for ontology. The terms are organized according to TFIDF. The terms with low TFIDF values are considered important. You can, of course, come up your own terms and include them in the ontology. However you should try to use the terms in the document first.
   - TFIDF: terms with low values are important.

** Use the next blank page to draw your final ontology. **Draw your tentative ontology first on a blank sheet of paper. Then, redraw your final version of the ontology on the next page.**

** Please measure the **time** from now to the time when you finish the ontology creation task.

** After completing the ontology creation task, please fill out the questionnaire.

TFIDF: terms with low values are important

Terms	TFIDF
shark	1.3
research	1.6
conserve	1.6
information	1.8
fish	2.0
institution	2.1
ocean	2.1
database	2.1
university	2.1
protect	2.2
dna	2.3
help	2.4
fin	2.6
trade	2.6
enforce	2.6
market	2.6
study	2.6
manage	2.7
people	2.7
law	2.7
administer	2.7
habitat	2.8
food	2.8
dealer	2.9
demand	2.9
biology	3.0
reproduce	3.0
data	3.0
laboratory	3.2
ecology	3.2
white shark	3.2
population	3.2
hammerhead	3.4
museum	3.6

QUESTIONNAIRE

1. It took me about ____ minutes from start to finish for ontology creation.

2. Did you use the attached document containing terms?

Yes ______ No ______

If you check *Yes*, please answer # 3. If you check *No,* please skip #3.

#		Very little /strongly Disagree - Very much/Strongly Agree						
3	The document was helpful in creating an ontology.	1	2	3	4	5	6	7
4	Ontology creation is mentally demanding.	1	2	3	4	5	6	7
5	Ontology creation is difficult.	1	2	3	4	5	6	7

6. Please indicate which part of the ontology creation task was difficult for you (e.g., web searching, term selection, term organization).

____

____

____

Experimental Instructions Given to the Control Group

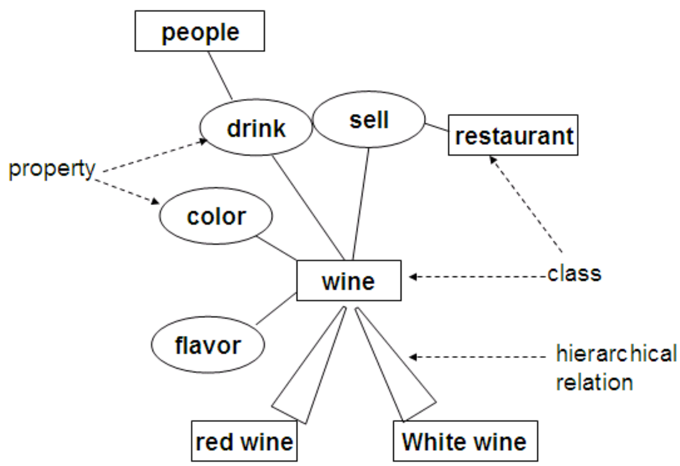
Please read the instructions and follow the tutorial presented below.

Instructions

You are going to develop an ontology about "**Shark and its Conservation**" which are **two main topics** for your ontology. For this exercise, 1) **search related web sites on "Shark and its Conservation" to find relevant terms**, then, 2) **create your mind map (also called an ontology) by following the tutorial** Below.

Tutorial for Ontology Creation

1. Determine important **terms** (can be either classes or properties) in ontology
   - `Example terms in Wine ontology: wine, color, flavor, sell, restaurant`
2. Define the **classes (rectangle)** from identified terms and the class hierarchy
   - `Example of wine subclasses: white wine, red wine`
3. Define the **properties (oval)** of classes
   - `Example of wine properties: color, flavor`



4. Classes can be linked (**line**) by shared property (example: sell)

** **Draw your tentative ontology first on a blank sheet of paper. Then, redraw your final version of the ontology on the next page.**

** Please measure the **time** from now to the time when you finish the ontology creation task.

** After completing the ontology creation task, please fill out the questionnaire.

QUESTIONNAIRE

1. The number of web sites I searched is ______ .

2. The number of web sites that have relevant information about sharks and their preservation is ______ .

3. It took me about ____ minutes from start to finish for ontology creation..

		Strongly Disagree						Strongly Agree
4	Ontology creation is mentally demanding.	1	2	3	4	5	6	7
5	Ontology creation is difficult.	1	2	3	4	5	6	7

6. Please indicate which part of ontology creation was difficult for you (e.g., web searching, term selection, term organization).

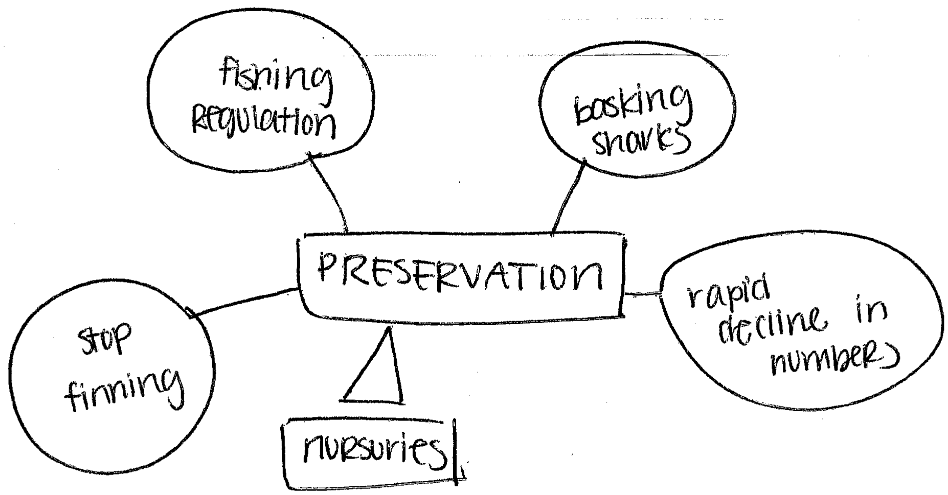
____

____

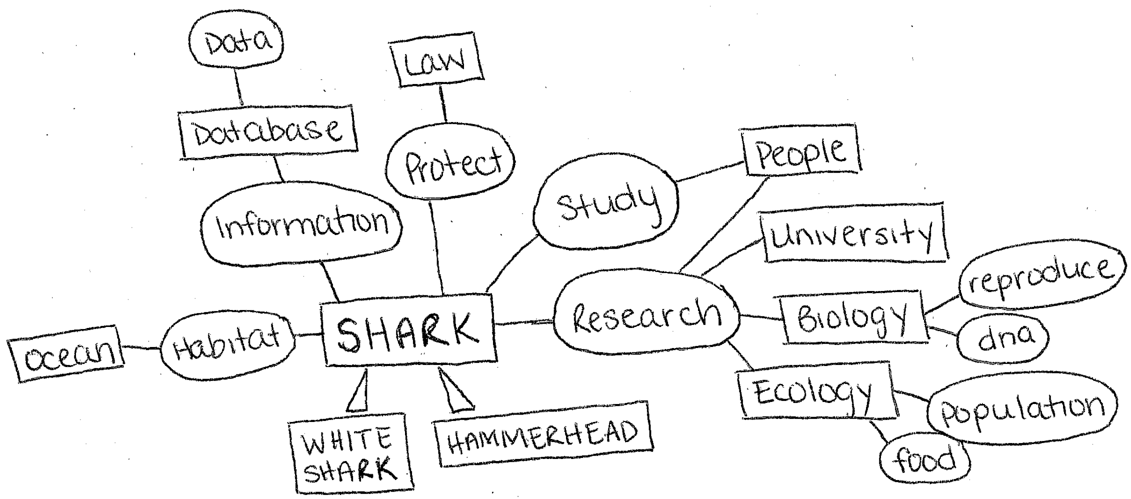
____

APPENDIX B

Examples of Ontologies and Evaluation



KC: 2, KR: 2, OQ: 1



KC: 6, KR: 7, OQ: 6

As shown in the figure above, the first figure receives low evaluation points (KC: 2, KR: 2, OQ: 1). KC is low because the number of relevant terms (fishing regulation, stop finning, and preservation) is only three. KR is also low. For example, nursery and basking sharks have little to do with sharks and their preservation. Finally as to OQ, the relationships of properties and inheritance are incorrect and irrelevant. As opposed to the previous example, the second figure received high evaluation points (KC: 6, KR: 7, OQ: 6). The total number of selected terms is 20, and the terms cover a large extent of the target ontology. The terms are very relevant to the ontology (e.g., habitat, protect, and DNA). Despite a minor mistake in identifying relevant properties (e.g., properties of biology), inheritance and relevant properties are correctly represented.

References

Banko, M., and Moore, R.C. "Part of speech tagging in context," Association for Computational Linguistics Morristown, NJ, USA, 2004.

Berners-Lee, T., Hendler, J., and Lassila, O. "The Semantic Web," *Scientific American* (5) 2001, pp 1-19.

Boley, D., Gini, M., Gross, R., Han, E.H., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., and Moore, J. "Document Categorization and Query Generation on the World Wide Web Using WebACE," *Artificial Intelligence Review* (13:5) 1999a, pp 365-391.

Boley, D., Gini, M., Gross, R., Han, E.H., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., and Moore, J. "Partitioning-based clustering for Web document categorization," *Decision Support Systems* (27:3) 1999b, pp 329-341.

Buitelaar, P., and Cimiano, P. *Ontology learning and population: bridging the gap between text and knowledge* Ios Pr Inc, 2008.

Burton-Jones, A., Storey, V.C., Sugumaran, V., and Ahluwalia, P. "A semiotic metrics suite for assessing the quality of ontologies," *Data & Knowledge Engineering* (55:1) 2005, pp 84-102.

Chakrabarti, S., Dom, B.E., Gibson, D., Kleinberg, J., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. "Mining the Link Structure of the World Wide Web," *IEEE Computer* (32:8) 1999, pp 60-67.

Corcho, O., Fernández-López, M., and Gómez-Pérez, A. "Methodologies, tools and languages for building ontologies. Where is their meeting point?," *Data & Knowledge Engineering* (46:1) 2003, pp 41-64.

Cristani, M., and Cuel, R. "A Survey on Ontology Creation Methodologies," *Int. J. Semantic Web Inf. Syst* (1:2) 2005, pp 49-69.

Dahlgren, K. "A Linguistic Ontology," *International Journal of Human-Computer Studies* (43:5-6) 1995, pp 809-818.

De Silva, L., and Jayaratne, L. "WikiOnto: A System for Semi-automatic Extraction and Modeling of Ontologies Using Wikipedia XML Corpus," IEEE, 2009, pp. 571-576.

Dennis, A.R., and Carte, T.A. "Using Geographical Information Systems for Decision Making: Extending Cognitive Fit Theory to Map-Based Presentations," *Information Systems Research* (9:2) 1998, pp 194-203.

Ding, Y., and Foo, S. "Ontology research and development. Part 1-a review of ontology generation," *Journal of information science* (28:2) 2002, p 123.

Dweck, C.S. "Motivational Processes Affecting Learning," *American Psychologist* (41:10) 1986, pp 1040-1048.

Fagni, T., Perego, R., Silvestri, F., and Orlando, S. "Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data," *ACM Transactions on Information Systems (TOIS)* (24:1) 2006, p 78.

Farquhar, A., Fikes, R., and Rice, J. "The ontolingua server: A tool for collaborative ontology construction," *International Journal of Human-Computers Studies* (46:6) 1997, pp 707-727.

Fensel, D., Van Harmelen, F., Klein, M., Akkermans, H., Broekstra, J., Fluff, C., Van Der Meer, J., Schnurr, H.P., Studer, R., and Hughes, J. "On-To-Knowledge: Ontology-based Tools for Knowledge Management," eBusiness and eWork 2000 Conf.(EMMSEC 2000), 2000.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. "METHONTOLOGY: From Ontological Art Towards Ontological Engineering," in: *Workshop on Ontological Engineering*, 1997.

Frantzi, K.T., and Ananiadou, S. "The C-value/NC-value domain independent method for multiword term extraction," *Journal of Natural Language Processing* (6:3) 1999, pp 145-180.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. "Sweetening Ontologies with DOLCE," Proceedings of EKAW, 2002, pp. 166-181.

Gruninger, M., and Fox, M.S. "Methodology for the Design and Evaluation of Ontologies," Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI, 1995.

Gu, T., Wang, X., Pung, H., and Zhang, D. "An ontology-based context model in intelligent environments," Citeseer, 2004.

Harrison, P.D., and Harrell, A. "Impact of" Adverse Selection" on Managers' Project Evaluation Decisions," *ACADEMY OF MANAGEMENT JOURNAL* (36) 1993, pp 635-635.

Hendler, J., Shadbolt, N., Hall, W., Berners-Lee, T., and Weitzner, D. "Web science: an interdisciplinary approach to understanding the web," *Communications of the ACM* (51:7) 2008, pp 60-69.

Herman, I. "Introduction to Semantic Web," in: *International Conference on Dublin Core and Metadata Applications*, Singapore, 2007.

Hong, W., Thong, J.Y.L., and Tam, K. "The Effects of Information Format and Shopping Task on Consumers' Online Shopping Behavior: A Cognitive Fit Perspective," *Journal of Management Information Systems* (21:3) 2004, pp 149-184.

Horrocks, I. "Ontologies and the Semantic Web," *Communications of the ACM* (51:12) 2008, pp 58-67.

Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E. "Ontology visualization methods - a survey," *ACM Computing Surveys* (39:4) 2007, pp 10:11-10:43.

Kietz, J.-U., Maedche, A., and Volz, R. "A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet," in: *Proceedings of EKAW-2000 Workshop*, 2000.

Lau, R., Song, D., Li, Y., Cheung, T., and Hao, J. "Towards a fuzzy domain ontology extraction method for adaptive e-learning,").

Lindland, O.I., Sindre, G., and Solvberg, A. "Understanding Quality in Conceptual Modeling," *IEEE Software* (11:2) 1994, pp 42-49.

Maedche, A., and Staab, S. "Discovering conceptual relations from text," Proceedings of ECAI-2000, 2000a, pp. 321-324.

Maedche, A., and Staab, S. "Semi-automatic engineering of ontologies from text," Proceedings of the 12th Internal Conference on Software and Knowledge Engineering, Chicago, USA., 2000b.

Maedche, A., and Volz, R. "The ontology extraction and maintenance framework text-to-onto," Proceedings of the ICDM '01 Workshop on Integrating Data Mining and Knowledge Management, California, USA, 2001.

Michener, W., Beach, J., Jones, M., Ludäscher, B., Pennington, D., Pereira, R., Rajasekar, A., and Schildhauer, M. "A knowledge environment for the biodiversity and ecological sciences," *Journal of Intelligent Information Systems* (29:1) 2007, pp 111-126.

Motik, B., Maedche, A., and Volz, R. "A Conceptual Modeling Approach for Semantics-Driven Enterprise Applications," in: *CoopIS/DOA/ODBASE,* R. Meersman and Z. Tari (eds.), Springer, 2002, pp. 1082--1099.

Noy, N.F., and McGuinness, D.L. "Ontology Development 101: A Guide to Creating Your First Ontology. 2001," *Standford Knowledge Systems Laboratory Technical Report KSL-01-05 and Standford Medical Informatics Technical Report SMI-2001* (880) 2001.

Omelayenko, B. "Learning of Ontologies for the Web: the Analysis of Existent approaches," Citeseer, 2001, pp. 97-106.

Paas, F. "Training strategies for attaining transfer of problem-solving skill in statistics: a cognitive-load approach," *Journal of educational psychology* (84:4) 1992, pp 429-434.

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

Ram, S., and Zhao, H. "Special Issue on Semantic Web: Opportunities and Challenges," *Information Technology and Management* (8:3) 2007, pp 203-204.

Salton, G., and Buckley, C. "Term-weighting approaches in automatic text retrieval," *Information Processing and Management: an International Journal* (24:5) 1988, pp 513-523.

Sánchez, D., and Moreno, A. "Learning non-taxonomic relationships from web documents for domain ontology construction," *Data & Knowledge Engineering* (64:3) 2008, pp 600-623.

Schreiber, G., Wielinga, B., and Jansweijer, W. "The KACTUS view on the 'O' Word," in: *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, 1995, pp. 159-168.

Shaft, T.M., and Vessey, I. "The Role of Cognitive Fit in the Relationship between Software Comprehension and Modification," *MIS Quarterly* (30:1) 2006, pp 29-55.

Sitkin, S.B., and Weingart, L.R. "Determinants of Risky Decision-Making Behavior: A Test of the Mediating Role of Risk Perceptions and Propensity," *ACADEMY OF MANAGEMENT JOURNAL* (38) 1995, pp 1573-1592.

Spyns, P., Meersman, R., and Jarrar, M. "Data modelling versus ontology engineering," *ACM SIGMOD Record* (31:4) 2002, pp 12-17.

Steinberg, E.R. "Cognition and learner control: a literature review, 1977-1988," *Journal of Computer Based Instruction* (16:4) 1989, pp 117-121.

Sugiura, N., Kurematsu, M., Fukuta, N., Izumi, N., and Yamaguchi, T. "A Domain Ontology Engineeering Tool with General Ontologies and Text Corpus," in: *Proceedings of the 2nd Workshop on Evaluation of Ontology based Tools*, 2003, pp. 71-82.

Suh, K., and Lee, Y.E. "The effects of virtual reality on consumer learning: an empirical investigation," *MIS Quarterly* (29:4) 2005, pp 673-697.

Tsikriktsis, N. "A review of techniques for treating missing data in OM survey research," *Journal of Operations Management* (24:1) 2005, pp 53-62.

Uschold, M., Artificial Intelligence Applications, I., and University of, E. *Building Ontologies: Towards a Unified Methodology* Artificial Intelligence Applications Institute, University of Edinburgh, 1996.

Vessey, I., and Galletta, D. "Cognitive fit: An empirical study of information acquisition," *Information Systems Research* (2:1) 1991, pp 63-84.
Xing, J., Han, M., and Tao, X. "A Wetland Protection Domain Ontology Construction for Knowledge Management and Information Sharing," *Human and Ecological Risk Assessment: An International Journal* (15:2) 2009, pp 298-315.
Zhou, L. "Ontology learning: state of the art and open issues," *Information Technology and Management* (8:3) 2007, pp 241-252.

Chapter 3

Using Pruning Methods to Query Bio-Ontologies

ABSTRACT

Researchers and professionals in bioinformatics have been developing large ontologies to organize knowledge in the field. As ontologies grow large, retrieving manageable amounts of information from them becomes a difficult and costly task. Our research applies a pruning approach based on cognitive load theory to help extract relevant aspects of knowledge from large bio-ontologies. A prototype called GOP (Gene Ontology Pruner) is developed by significantly extending prior research. This prototype supports systematic identification of relevant concepts and deletion of irrelevant parts of an ontology. To evaluate the usefulness of the pruning approach, an experiment based on the cognitive load theory is conducted. This study finds that information filtering, task complexity, cognitive fit significantly impact cognitive load which, in turn, affects task performance.

Keywords
Ontology, pruning, cognitive load, cognitive fit, information filtering, gene ontology

1. INTRODUCTION

The size of biological data has recently increased explosively with the arrival of new technologies and methods. For example, human genome research has churned out a great deal of data with the use of microarrays and massively parallel signature sequencing (Oudes et al. 2005; Zhang et al. 2004). However, making this enormous amount of knowledge sharable and reusable is complex and difficult. Recent research has recognized the potential of ontologies in improving the ability to share and reuse knowledge in complex domains such as biological sciences (Baker 1999; Lord et al. 2003; Mizoguchi et al. 1995; Stevens et al. 2001; Yeh et al. 2003).

Ontology is an explicit specification of a conceptualization (Gruber 1993). A conceptualization is an abstract, simplified view of the world that we want to represent. Domain ontologies are formal descriptions of the classes of concepts and the relationships among those concepts that describe an application area (Musen 1998). The use of ontologies in complex domains such as bioinformatics has several benefits: data integration, information retrieval, facilitation of knowledge sharing and reuse, improved interoperability of systems, maintenance, and reliability. Ontology can be used for the integration of data and to improve the effectiveness of queries used to access information (Blake and Bult 2006). Ontology as a mechanism to specify consensus enables users to share and reuse knowledge across different applications and stakeholders (Borst 1997; Holsapple and Joshi 2002). As ontology provides a common terminology over a domain, it provides the foundation for interoperability between information systems. It can be used as an index to a repository of information. For example, an ontology in the field of software maintenance has structured and generalized information that facilitates reuse of information, which can improve maintenance quality while decreasing maintenance cost (Ruiz et al. 2004). Similarly, an ontology can serve as a foundation for checking design consistency against specifications (Uschold and Gruninger 2009).

As the size of an ontology grows, the ability to effectively use it is severely restricted. For example, since the amount of data collected in the field of biology has exploded in the recent years, the size and scope of bio-ontologies that have been created to support this field also have increased exponentially. Currently the Gene Ontology (GO), a well-known bio-ontology, has more than 19,000 entities, and it continues to grow in size. The current rate of growth of bio-ontologies suggests that they will become very large, as has been the case with ontologies in the medical domain. For example, UMLS, a popular bio-ontology, contains more than one million concepts. In addition, several bio-ontologies are being integrated into large ontologies such as GO (Blake and Bult 2006). Therefore, the use of methods to access only relevant aspects of knowledge contained in ontologies will be critical for their effective use. Ontology pruning is increasingly recognized as a promising approach for this purpose (Good et al. 2006). Motivated by this need, we investigate the following research question: *1) How can relevant aspects of knowledge be extracted from large ontologies to reduce the effort involved in their effective use?*

In addition, our research also studies the impact of two important factors that affect task performance, viz., the format used to represent the ontology and the complexity of the task. Prior research has suggested that the fit between the problem representation and the task significantly affects task performance (Vessey 1991). Task complexity has been shown to impact human cognitive processes, which in turn affects performance. Therefore, our study includes two additional research questions: *2) How does the format in which ontology information is presented affect task performance? 3) How does the complexity of the task affect task performance?*

To answer the research question, our research approach uses the following steps: 1) develop an appropriate method for pruning ontologies, 2) develop a prototype to demonstrate the feasibility and applicability of the method to pruning ontologies, and 3) evaluate the effectiveness of the proposed approach. In Section 2, we discuss the theoretical background on ontology pruning. Here, we compare several pruning methods by identifying their strengths and weaknesses. We propose an ontology pruning approach by significantly extending an existing pruning approach. In section 3, we present the architecture of a prototype system that helps prune ontologies. The capabilities of the system are presented and compared with those of AmiGO, a popular query tool which is integrated with a popular ontology. In the section 4, a theoretical model for the evaluation of the effectiveness of our approach to ontology pruning is introduced. In the section 5, the research method and design used in the empirical evaluation are discussed. In the section 6, the results of data analysis are presented with limitations of the experiment. In the section 7, discussion and implications of this research are presented. It is followed by a discussion of theoretical and practical contributions of this research.

2. PRUNING BIO-ONTOLOGIES

When an ontology becomes very large, it may not be possible to use it efficiently and effectively. Retrieving relevant information from large ontologies is often very difficult. For example, when the GO ontology, a popular ontology used in bioinformatics, is queried with keywords using AmiGO, a browser and search engine tool, typically several hundred matching results are returned. Unfortunately, such results often overwhelm the user with a number of results that are not of relevance to the user. Ontology pruning is an approach to retrieving relevant ontologies from large ontologies by removing irrelevant concepts (Navigli 2002). When the elements retrieved from an ontology are treated as representing a conceptual schema of the domain that are relevant to the user, pruning may be viewed as a method for creating concise conceptual schemas with highly relevant components.

A generic pruning task consists of two phases shown in Figure 1. They are the selection phase and the pruning phase. The selection phase identifies elements relevant to the domain. A wide variety of selection methods have been proposed in the literature. Some methods use text processing techniques to select relevant elements (Buitelaar et al. 2006; Maedche and Volz 2001), while others require that the user select the relevant documents manually (Bhatt et al. 2004). For example, say a researcher in the field of bioinformatics is interested in identifying all biological processes related to TRAF6, a gene associated with protein signal transducer. An ontology may help the researcher identify two biological processes (positive regulation of the IkappaB kinase/NF-kappaB cascade, and positive regulation of T cell cytokine production) that are related to TRAF6. The identification of these processes corresponds to the *selection phase*.

The pruning phase utilizes the information attained from the selection phase to remove irrelevant or useless elements by taking into account the characteristics of the target domain. For example, based on the two biological processes that were selected, a pruning method first identifies all related concepts, such as biological processes, cellular components and molecular functions. Since the size of collected processes is typically very large, a pruning method removes irrelevant concepts and creates a pruned ontology.

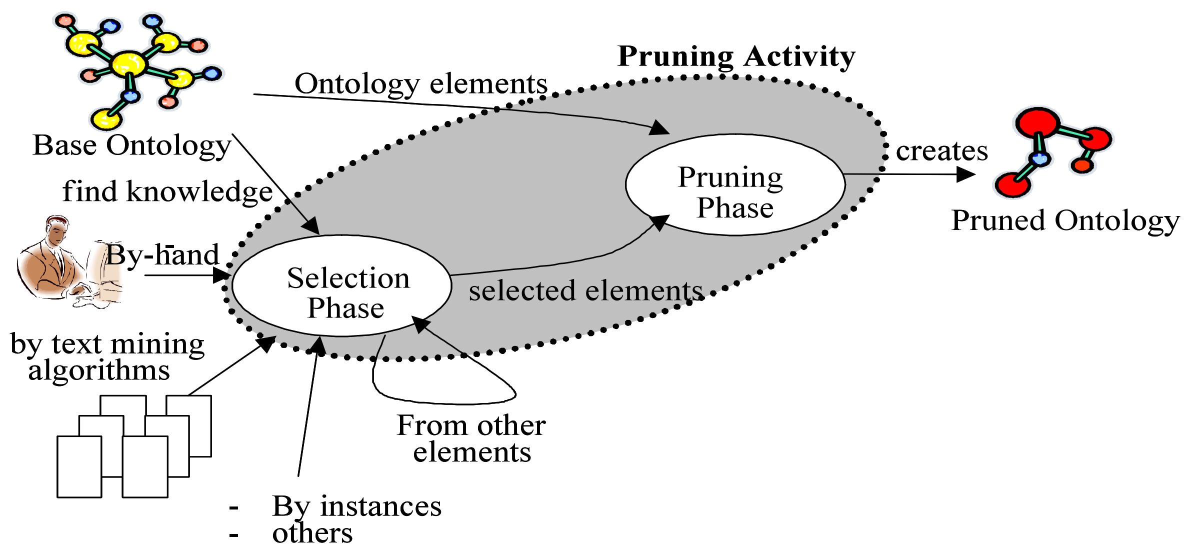


Figure 1. Steps in ontology pruning

2.1 Pruning Methods

We use the GO ontology in our study of ontology pruning methods. GO is one of the largest ontologies in biology. It has 19861 terms that include 10690 biological processes (e.g. electron transport, gluconeogenesis), 1740 cellular components (e.g. inner membrane, cytoplasm), and 7431 molecular functions (e.g. monooxygenase activity). AmiGO, the tool that provides an interface to GO, allows users to use keywords to find matching terms. Thus, AmiGo supports only the selection phase.

We evaluated the three pruning methods developed by prior research. These methods were selected because of their suitability for semi-automatic ontology pruning. In this section, we discuss their goals, pruning approach, and assess their strengths and weaknesses.

2.1.1 Knowledge Bus

The pruning method used in the Knowledge Bus system (Peterson et al. 1998) deletes irrelevant elements in an ontology. Although this pruning method was developed for pruning very large ontologies, it typically deletes only a small number of elements because the goal is to delete only elements that are not directly related to the concept of interest. In the Knowledge Bus system, a concept C is related to a relevant concept C_T if we can find a sequence of relationship types that connect C to C_T, where a relationship type may be either taxonomic (*IsA* relationships) or non-taxonomic (called "associations" in Unified Modeling Language (OMG 2003)). The goal of this pruning method is to obtain a pruned ontology that contains the relevant elements as well as other elements that are directly or indirectly related to them. This method is very generous in that it does not delete any concept which is related through a super-type relationship or a non-taxonomic relationship to the concept of interest. As a result, Knowledge Bus is not suitable to retrieve a compact ontology from a very large ontology such as GO. In addition, compatibility with Web Ontology Language (OWL) is important in pruning because OWL has become a standard XML language for authoring ontology since 2002. Knowledge Bus which was developed in 1998 is not compatible with OWL.

2.1.2 Swartout method

Swartout et al. (1996) developed and applied pruning to a linguistic ontology called SENSUS. The SENSUS ontology contains over 70,000 concepts. The pruning method was used to develop an ontology for the military air campaign domain from SENSUS.

In the pruning process of Swartout method, users select seed terms. The pruning starts from selecting all parent concepts of the seed terms. Then, parent concepts with many relevant subtype concepts are selected because they are considered important. Finally, branch concepts of all parent concepts (including the concepts considered irrelevant in the beginning) are added because these are now considered important as subtype concepts of important parent concepts.

Ontology pruned by Swartout method tends to be large when it is applied to a huge ontology. Therefore Swartout is not suitable to prune GO. Also, Swartout method does not handle ontologies authored in OWL.

2.1.3 Conesa and Olive method

This pruning method (Conesa and Olivé 2004) has been designed to create conceptual schemas of information systems from general ontologies. It is composed of two phases: the first phase selects the concepts that are relevant for the information system. These concepts are called concepts of direct interest (*CoI*). The selection of the CoI concepts can be done by using several strategies, such as using the requirements of the information system, querying the user, using text-mining algorithms, etc. Therefore, the first phase can be either manual or automatic.

In the second phase, unnecessary elements such as irrelevant concepts and parents concepts are deleted in three steps: 1. Pruning irrelevant concepts and constraints, 2. Pruning unnecessary parents, 3. Pruning unnecessary generalization paths. This pruning method generates compact and small ontologies and with very few irrelevant concepts. The second phase can be executed automatically.

Conesa and Olive method provides a compact result when it is applied to a large ontology such as GO. However, it is not compatible with OWL. In addition, it cannot handle instances of classes (e.g. Holiday Inn of Hotel).

2.2 Comparison

A case study was conducted to exemplify the different pruning methods and to compare their benefits and drawbacks. Consider a user who is interested in biological processes associated with viral perturbations of host cells. The user searches for relevant biological processes based on eight selected biological processes (*viral perturbation of host cells, regulation of translation, cellular bio-synthesis, cellular macromodule metabolism, protein metabolism, cellular metabolism, viral life cycle and metabolism*) out of 41 biological processes. Three pruning methods are evaluated based on this case study.

Table 1. Comparison of the main current pruning methods

	Base ontology	Integrity Constraints	Selection Strategy	Size of the Pruned Ontology	Compatible with OWL	Instance Handling
Knowledge Bus	*Cyc*	Yes	manual	very large (36/41)	Not compatible	Not applicable
Swartout et al.	*SENSUS*	No	manual	very large (37/41)	Not compatible	Not applicable
Conesa and Olive	ResearchCyc,	Yes	Either manual or automated	Small (11/41)	Not compatible	Not applicable

Table 1 summarizes several characteristics of the pruning methods and results from the case study. These include: 1) the base ontology used; 2) whether or not the method takes into account the integrity constraints that specify the conditions that the instances of the ontology must fulfill in order to be correct (for example, a condition that states that a professor must have a PhD); 3) how automated the method is; 4) the strategy used for selecting concepts of direct interest[1]; and finally, 5) the efficiency of the pruning activity measured by the number of elements in the pruned ontology.

Pruning methods consider a particular context for their application. The ontology context determines its foremost properties: 1) the base ontology the method is able to prune, 2) how the method selects concepts of direct interest, and 3) how many elements are pruned. For example, pruning methods that support information systems development help prune expressive ontologies, which have more axioms, such as relationship types and rules, compared with less expressive ontologies. The rationale behind this approach is that the user knows all the concepts. Swartout et al method uses linguistic ontologies as a base ontology, which is less expressive than those for Knowledge Bus, and Conesa and Olive methods. But its base ontologies may contain more concepts than the others do. For example, SENSUS ontology used by Swartout, et al., has more than 50,000 concepts, while OpenCyc (the ontology pruned by Conesa and Olive) has less than 5,000 concepts. These methods have more efficient selection processes, because they use the linguistic relationships (synonyms, antonyms, etc) among concepts in the linguistic ontologies. Swartout method tends to generate large pruned ontologies, because the pruned ontologies are used for programs to infer information, and contain concepts of direct interest and all related concepts. For this reason, this pruning method is similar to the first step in the Conesa and Olive method.

Each pruning method works well in specific contexts. Knowledge Bus is very useful to identify information related to a given concept. The Swartout method helps the user learn about

[1] A concept is of direct interest in a given ontology when the ontology users and designers are interested in either representing its population or inferring new information from it. It is denoted as CoI

the selected domain, because of the interactive identification of new concepts related to the seed concepts which are manually selected by the user in the selection phase. Conesa and Olive method has been used to prune ontologies without instances such as UML ontologies. When an ontology contains instances, this method may leave the ontology in an inconsistent state because the classifiers of the instances can be deleted during the pruning activity while the instances are not deleted. Since Gene Ontology contains instances and is recorded in OWL, Conesa and Olive method is not directly applicable.

2.3 Our approach

Conceptually, the pruning process may be explained as follows: The concepts not included in this selection are deleted in the first step. Not all the parents of the queried concepts are necessary, and only the ones required to keep the inheritance relations between CoI elements are necessary. Hence, the non-CoI elements that have noCoI concept as supertype are deleted. Finally, the redundant inheritance paths between relevant elements and the orphan individuals are deleted.

The pruning approach developed in this research is designed to effectively prune an ontology with instances and handle ontologies represented in OWL. The execution of the method is composed of two stages, viz., the selection process and the pruning process. In the selection process the elements that are relevant to the constraints, attributes and relationships inherited by *CoI* concepts, all the *CoI* supertypes are selected in the set *G(CoI)*.

$$G(CoI) = \{c \mid c \in CoI \vee \exists sub\ (\mathrm{IsA}^{+}(sub,c) \wedge sub \in CoI)\}^{2}$$

[2] We denote by $IsA(C_1,C_2)$ the generalization relationship (inheritance) between concepts C_1 and C_2. IsA^{+} will be the transitive closure of *IsA*

We call constrained concepts of an integrity constraint *ic*, $CC(ic)$, the set of concepts appearing in the formal expression of *ic*. $CC(O)$ is to denote the set of concepts constrained by all the integrity constraints defined in ontology *O*. Formally,

$$CC(O) = \{c \mid c \text{ is a concept} \wedge c \in O \wedge \exists ic\ (ic \text{ is a constraint} \wedge ic \in O \wedge c \in CC(ic))\}$$

After the selection stage, selected information is used within the pruning stage. The pruning module interacts and prunes the ontology in four phases:

1) Pruning irrelevant concepts and constraints: the elements of the ontology that have not been selected in the selection module are deleted. The concepts and constraints to delete are denoted by the following sets:

$$\text{IrrelevantConcepts} = \{c \mid c \text{ is a concept} \wedge c \in O_0 \wedge c \notin G(CoI)\}$$

$$\text{IrrelevantConstraints} =$$

$$\{ic \mid ic \text{ is a constraint} \wedge ic \in O_0 \wedge \exists c\ (c \in CC(ic) \wedge c \notin G(CoI)\}$$

2) Pruning unnecessary parents: After the previous step, the concepts of the resulting ontology (O_1) are exactly *G(CoI)*. However, the concepts strictly needed are given by:

$$NeededConcepts = CoI \cup CC(O_1)$$

The other concepts (i.e. those given by $G(CoI) - NeededConcepts$) are potentially not needed. We can prune the parents of *NeededConcepts* which are not children of some concept in *NeededConcepts*. Formally,

UnnecessaryParents =

$$\{c \mid c \notin \text{NeededConcepts} \wedge \neg\, \exists c'\ (c' \in \text{NeededConcepts} \wedge \text{IsA}^{+}(c,c'))\}$$

The result of this step is the ontology O_2:

$$O_2 = O_1 - \text{UnnecessaryParents}$$

3) Pruning unnecessary generalization paths: it deletes the elements that belong to redundant generalizations.

   A generalization path exists between C_1 and C_n if:

   - C_1 and C_n are two concepts from O_2,
   - $IsA^+(C_1,C_n)$ and
   - The path includes two or more generalization relationships $IsA(C_1,C_2)$, …, $IsA(C_{n-1},C_n)$.

   A generalization path $IsA(C_1,C_2)$, …, $IsA(C_{n-1},C_n)$ between C_1 and C_n is potentially redundant if none of the intermediate concepts C_2, …, C_{n-1}:

   - Is member of the set $CoI \cup CC(O_2)$
   - Is the super or the sub of other generalization relationships.

   A generalization path between concepts C_1 and C_n is redundant if there are other generalization paths between the same pair of concepts. In this case, we prune the concepts C_2, …, C_{n-1} and all generalization relationships in which they participate.

   The output of this step is the ontology, O_3.

4) Pruning orphan individuals: it deletes the instances (orphan individuals) whose classifiers have been deleted. After the previous steps have pruned the concepts of the ontology, the individuals of the ontology must be pruned as well. This step removes the instances of the ontology such that all its classifiers have been deleted in the previous steps. When an instance is deleted, all its value properties and sameAs relationships are deleted as well. The set of instances to delete are formally:

OrphanIndividuals =

$$\{i \mid i \text{ is an individual} \wedge i \in O_0 \wedge \neg \exists c\ (c \in O_3 \wedge InstanceOf(i,c))\}$$

The result of this step is the pruned ontology O_P:

$$O_P = O_3 - OrphanIndividuals$$

The fourth step in the pruning processes explained above deletes the instances of the ontology that have become orphan because its classifiers have been deleted in the previous steps. This step is necessary for pruning ontologies such GO which are represented in OWL and include instances.

In the following section, the architecture and features of a prototype called Gene Ontology Pruner (GOP) in which we incorporate this method are presented.

3. GENE ONTOLOGY PRUNER

3.1 Implementation

The GOP prototype has been implemented as a web application using Java Server Pages (JSP). This development environment was chosen because it would make the system easily accessible through the Web. On the client side, a web browser is used to gather relevant terms from the user. On the server side, several modules have been implemented using java servlets. The purpose of these modules includes parsing the query, identifying the GO concepts relevant to the query, pruning the ontology, saving it as an OWL (Web Ontology Language) file and creating a graphical representation of the pruned ontology. The selection and pruning modules interact with Gene Ontology through two different Java APIs. They interact with the OWL version of Gene Ontology through the OWL and OWL-S APIs (Ashburner et al. 2000), which are used for accessing and pruning the concepts of the ontology as well as for generating the pruned ontology.

3.2 System Architecture

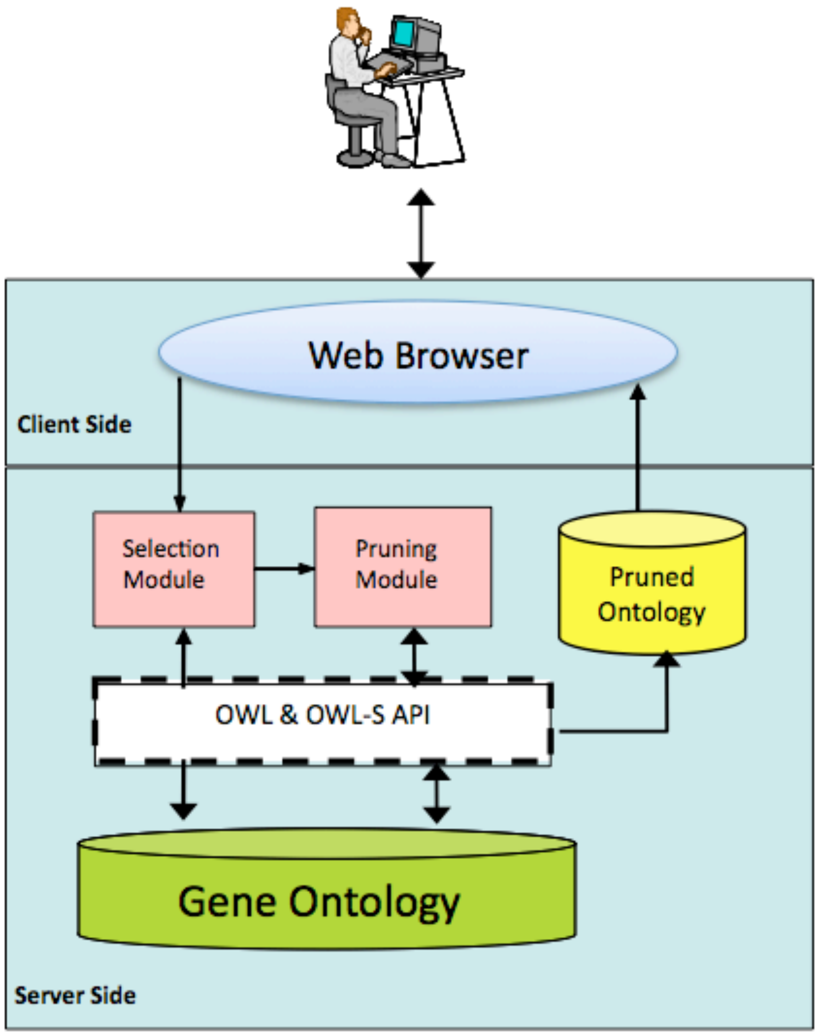


Figure 2. System Architecture

GOP is developed to support the development of an ontology by pruning GO. It allows users to select multiple biological terms in order to retrieve the most relevant terms. Figure 2 shows the architecture of GOP which consists of two parts: the client side and the server side. A web browser based interface is used to gather information from users and present the query results provided by the server. The server side has two major modules: selection module and pruning module.

The selection module captures the user's query (or concepts of interest (CoI) to the user) and retrieves related concepts from GO. Specifically, all supertypes of the query element are retrieved. This result is delivered to the pruning module. OWL & OWL-S API support the interactions among the selection module, pruning module, and GO.

The pruning module interacts with and prunes GO in four phases as explained in the previous section. The pruned ontology is stored in the server, and a webpage containing the taxonomy of the pruned ontology is presented to the user.

3.3 Sample Query

In this section, the functionality of GOP is presented using an example. Suppose a researcher in bioinformatics is interested in the proteins used in biosynthesis and cellular metabolisms. S/he submits a query that contains the following terms: *macromolecule biosynthesis* (GO_0009059), *regulation of cellular metabolism* (GO_0031323) and *regulation of protein biosynthesis* (GO_0006417). With the AmiGO tool, each of these terms must be submitted separately to retrieve relevant concepts from GO. In contrast, GOP accepts queries that include multiple terms.

The user poses the query through the web interface provided by GOP (see figure 3). After the query is sent to the server, the selection module identifies the Gene Ontology elements that correspond to the queried concepts (CoI concepts) GO_0009059, GO_0006417 and GO_0031323 and all their supertypes. After the selection and pruning phases are completed, a new webpage is created and sent to the user (see figure 4). This webpage contains the taxonomy of concepts of the pruned ontology and a link to the location where the pruned ontology can be downloaded. Three underlined concepts are those selected at the selection phase. The user can view the query results or download the pruned ontology as shown at Figure 4. The concepts selected during the selection phase are underlined in red in Figure 4..

Figure 3. Query interface of GOP

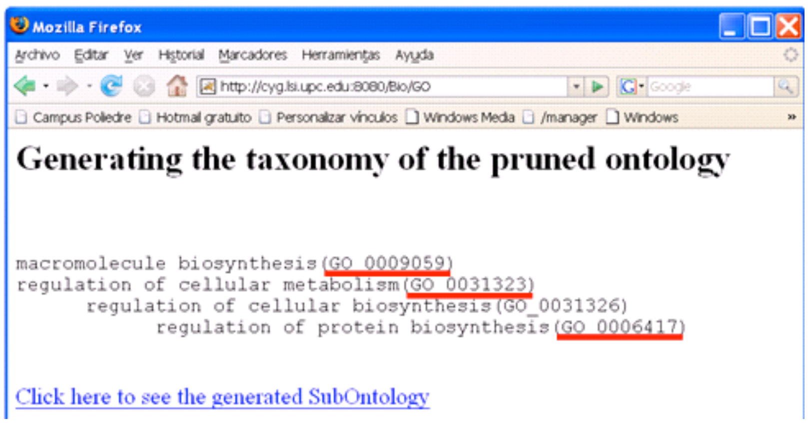


Figure 4: Query results

3.4 Comparison with AmiGO

AmiGO is the most popular tool used to query Gene Ontology (http://www.genedb.org/amigo/perl/go.cgi). It graphically displays the query results. However, it suffers from two problems that are addressed in the development of GOP: 1) it does not include any pruning capability and therefore, the retrieved results often include a large number of irrelevant concepts, and 2) the user query can contain only one term.

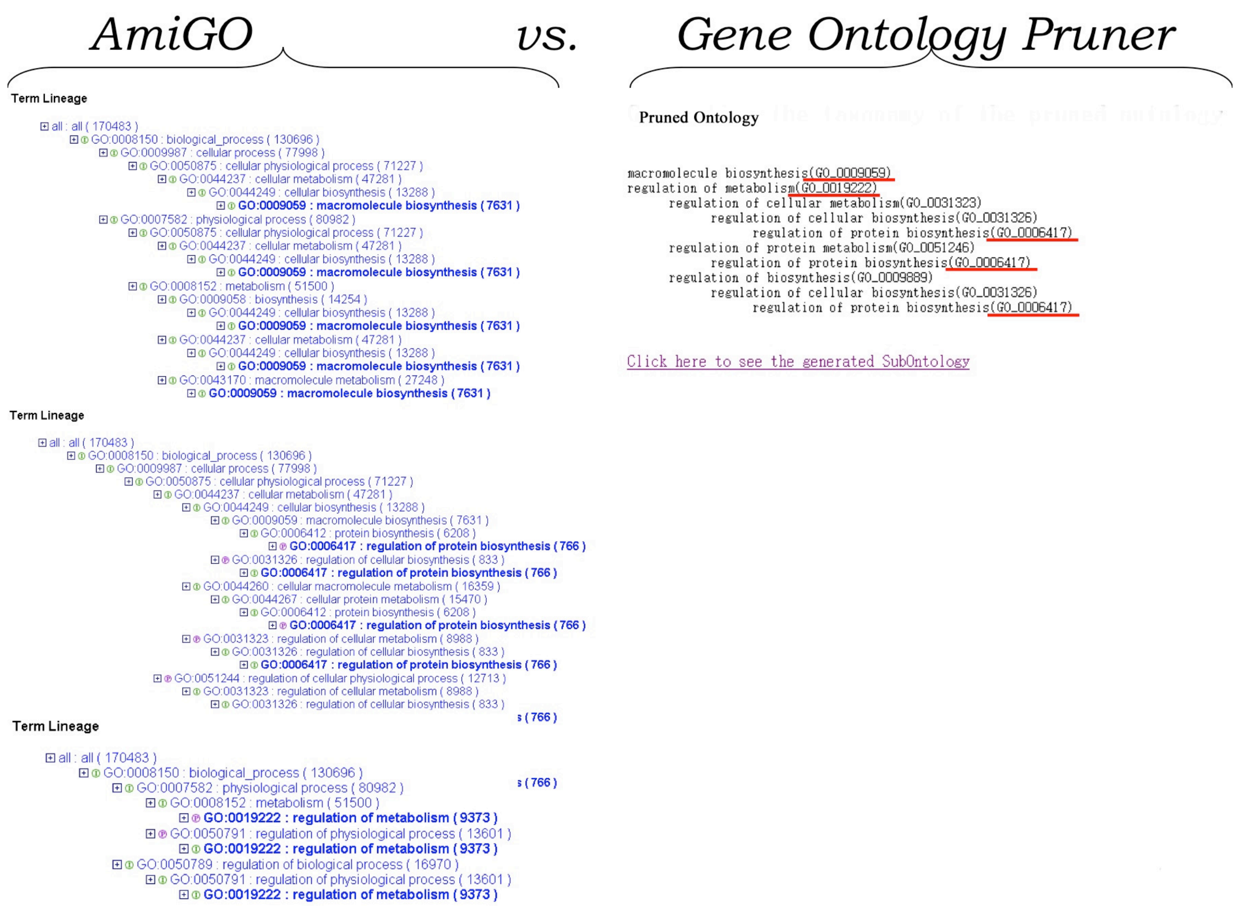


Figure 5: Outputs from AmiGO and GOP for the same query.

Suppose the user is interested in identifying three concepts: macromodule biosynthesis, regulation of metabolism, and regulation of protein biosynthesis. In AmiGO the user needs to perform three queries, one for each term. The left side of Figure 5 shows the results generated by AmiGO for each of these terms. It shows partial results from the search using each of the three concepts, which are shown in bold. The results for each search term is several pages long. Thereafter, the user must study the result of three queries together to identify possible relationships among them. However, with GOP, the user needs to perform only one search (see the output from GOP in the right side of Figure 5). Since the output from AmiGO spans several pages only partial result for each search is shown in Figure 5. However, the complete results from GOP are displayed because they are concise and take into consideration the relationships among all the three concepts.

By default, AMIGO delivers search results in text format. Also, users can choose to view the results in a graphical format. However, the effect of the two formats on task performance in the retrieval of ontologies has not been examined in prior research. In the next section, we present a study which empirically evaluates whether the ontology pruning approach implemented in GOP provides relevant retrieval results.

4. RESEARCH MODEL FOR THE EVALUATION OF ONTOLOGY PRUNING

Cognitive load theory can be used to explain why ontology pruning may improve task performance. Cognitive load refers to the load on the working memory during problem solving, thinking and reasoning. Causal factors of cognitive load can include amount/format of information, the task, characteristics of subjects, the environment, and their mutual relations (Kirschner 2002a). Cognitive load may be classified as: intrinsic cognitive load, extraneous cognitive load, and germane cognitive load (Sweller 1988; Sweller and Chandler 1994). Intrinsic cognitive load is determined by the interaction between the nature of material to be learned and learner's expertise. Extraneous cognitive load is the extra load beyond the intrinsic cognitive load resulting from poorly designed instruction, whereas germane cognitive load is the load related to processes that contribute to the construction and automation of schemas (Paas et al. 2003). Ontology pruning helps reduce intrinsic cognitive load because the level of intrinsic cognitive load depends on the nature of the material to be learned and the amount of information processing needed. In other words, the higher the number of elements that must be processed simultaneously, higher the level of intrinsic cognitive load (van Merriëboer and Sweller 2005).

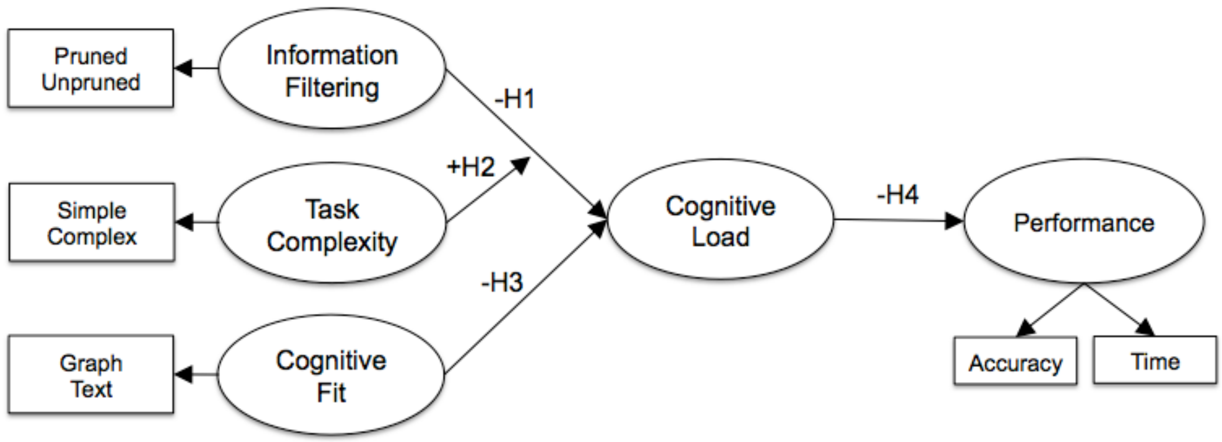


Figure 6: Research Model

Information filtering is a method for pruning irrelevant information (Malone et al. 1987). It is different from information retrieval, in that information filtering involves the process of removing irrelevant information, while information retrieval involves the process of finding information (Foltz and Dumais 1992). Information filtering reduces cognitive load by selecting relevant parts from a larger set of information and presenting it in a prioritized order (Malone et al. 1987). For example, wearable computers can use the wearer's current state to filter out unnecessary information, thereby reducing cognitive load (Billinghurst and Starner 1999). Due to the limited capacity of human to store current information in memory, people filter out information when they have to process a lot of information (Broadbent 1958). People tend to generate and respond to simpler information when they are overloaded with large amounts of information (Jones et al. 2004). The problem even more pronounced when unnecessary information is included in a large volume of information presented to the user. Therefore, in order to reduce the cognitive load, it is important to reduce the amount of information presented to the user.

Ontology pruning is a form of information filtering in that it identifies parts of a large ontology that are relevant to a specific context. The goal of pruning is to automatically extract the subset of the relevant conceptualization to the target domain (Volz et al. 2003). Domain ontology can be created by applying ontology pruning methods to a large ontology. Domain ontology is an ontology for a specific application domain (e.g. airline reservation, healthcare). Prior research has focused on identifying relevant documents to design domain ontologies depending on different situations and source ontologies (Maedche and Staab 2000).

Ontology pruning as a form of information filtering focuses on automatic extraction from a large ontology. Ontology development and management require a lot of time and effort. In particular, manually retrieving relevant information from a large ontology takes significant mental effort. Prior research suggests that ontology pruning method may reduce this mental effort (Maedche and Volz 2001). Therefore, we hypothesize that ontology pruning will reduce cognitive load.

Hypothesis 1: Information filtering provided by ontology pruning will negatively affect cognitive load.

Prior research in decision making suggests that decision makers experience higher cognitive load as task complexity increases (Johnson and Payne 1985). Task complexity has been studied using three perspectives (Campbell 1988): 1) complexity as primarily a psychological experience, 2) complexity as an interaction between task and person characteristics, and 3) complexity as a function of objective task characteristics. This research subscribes to the third perspective in which task complexity can be measured objectively.

This study focuses on the moderating effect of task complexity on the relationship between information filtering and cognitive load. Both Campbell (Campbell 1988) and Wood (Wood 1986) note that complex tasks are characterized by a lot of information processing, high uncertainty, and many alternatives/paths. Knowledge intensive tasks such as making sense of biological processes involves significant amount of information. Simple tasks typically have low information load. A low level of interaction between information and cognitive load would be needed to perform simple tasks (Nickerson and Zenger 2004). In simple tasks, filtered information has limited impact on cognitive load. On the other hand, in complex tasks , task complexity will influence the relationship between information filtering and congnitive load. Complex tasks would heighten the negative relationship between filtered information and cognitive load. In complex tasks, task performer will feel less cognitive load with well-filtered information.

Thus, we hypothesize,

Hypothesis 2: Task complexity will moderate the relationship between information filtering and cognitive load

Information representation format provided by information systems can affect user's decision making. Users of geographical information systems (GIS) can make a faster decision based on graphical information when compared with table-based information (Dennis and Carte 1998). A major use of the bio-ontology involves the retrieval of information on genes and biological processes and understanding the relationships among them. In particular, biological processes are structured in a hierarchy. Understanding the functional relationships among biological processes may be better supported by information in a graphical format rather than as text. Users can be better supported by graphic-based information in understanding the structural relationships among biological processes.

In order to evaluate the effects of two types of representations of the pruned ontology, viz., graphical vs. textual on cognitive load, we use the Cognitive Fit Theory (CFT). The CFT explains how information representation affects the decision processes and outcomes of decision making. According to CFT, decision makers develop a mental representation of the task and adopt decision processes based upon the task and presentation of task information. Vessey (1991) argues that decision makers can deliver faster and more accurate solutions when the presented information matches the mental representation of the task. This is because they use the same mental representation and decision processes for both the representation and the task. Thus, the fit between information presentation, task, and decision processes may affect performance.

Information in graphic format may significantly reduce the cognitive load in understanding the relationships among biological processes because it may better fit user's cognitive representation for a task which requires information to be presented in a structured format (Vessey 1991). Therefore, we assume that information in graphic format provides higher level of cognitive fit in task involving the understanding of relationships among biological processes. Thus, we hypothesize that high level of cognitive fit provided by information in graphic format reduces cognitive load.

Hypothesis 3: Cognitive fit will negatively affect cognitive load

Task performance is defined as the effectiveness with which people perform activities that contribute to the organization's technical core either directly by implementing a part of its technological process, or indirectly by providing it with needed materials or services (Borman and Motowidlo 1993). In biology, structural knowledge of biological entities and processes plays an important role in increasing effectiveness of task performance. Since the amount of knowledge available has increased exponentially with an explosive growth of biological data, it is very difficult (if not impossible) for human to process this knowledge.

Bio-ontology is used to form a semantic framework for data storage, retrieval and analysis. One of the typical tasks for researchers in bioinformatics using GO is to map biological information in a semantic framework. Many predictions and interpretations of data in biology are made by comparing the data in hand against existing knowledge (Baker et al. 1999). For example, biologists predict the structure of proteins from amino acid sequences using knowledge of known protein structures and examining that can sensibly represent the structure of the unknown protein.

Performance of tasks that involve the use of ontological information may be measured in terms of the effectiveness and efficiency of knowledge base creation. Efficiency is defined as the effort needed for task performance, and effectiveness is defined as the accuracy with which the task is performed (Moody 2004). For example, decision time and accuracy can be used to measure efficiency and effectiveness in the use of a Geographic Information Systems (Dennis and Carte 1998). Our study measures both time and accuracy to represent task performance.

An interdisciplinary study (Ramanujan et al. 2000) has shown that cognitive load resulting from psychological complexity negatively affects performance in software maintenance tasks. Information search in e-commerce applications is completed quickly when cognitive load is low (Rowley 2000). Similarly, as an ontology becomes large and complex, the cognitive load increases and therefore it negatively affects task performance unless unrelated information is removed (Kirschner 2002b).

Prior research has suggested that the relationship between cognitive load and task performance is complex. Task performance can address an increase of cognitive load from a task by putting more mental effort within the limits of their cognitive capacity (Paas et al. 2003). Mental effort can be treated as an indicator of cognitive load. A task performer can increase cognitive load to address complex tasks. The positive relationship between cognitive load and task performance is found when the task is simple (Paas et al. 2003). When cognitive load is high enough to exceed their cognitive capacity, decision makers cannot process information cues properly and therefore, cognitive load negatively affects task performance

(Denis and Carte 1998). Therefore, we hypothesize that high-level cognitive load reduces task performance.

Hypothesis 4: Increased cognitive load will negatively affect task performance.

5. RESEARCH METHOD AND DESIGN

5.1 Procedure and Subjects

A laboratory experiment was conducted to test the causal relationships between constructs in the research model. The experiment involves a 2 X 2 mixed factorial design with two exogenous variables, information filtering and cognitive fit, each of which were manipulated at two levels. All subjects in four cells received both simple and complex tasks. Pilot tests were conducted to refine the treatments and validate the measures. Undergraduate students at a large southeastern university in U.S. with background in information systems and biology served as subjects in this study. A total of 128 subjects participated in the experiment. Their mean of job experience in relevant fields was 2.4 years. Forty six percent of the subjects were male, and 54 percent were female.

Randomly selected subjects received either pruned or unpruned biological processes information related to biological genes such as TBK 1 and TRAF 6. Both TBK 1 and TRAF 6 were used in the experiment because they are important genes associated with several biological processes of virus infection. The biological processes information was prepared in two formats: graphical and text. The experimental procedure consisted of two parts. In the first part, each subject received information on the task, related biological genes such as TBK 1 and TRAF 6, and ontology. In the second part, subjects were asked to complete a questionnaire and measure the time taken to finish the experimental task. Both simple and complex tasks were given to subjects. The simple task involved answering five questions about the relationships among ten biological processes related to TBK 1. The complex task involves answering nine questions about the relationships among twenty-nine biological processes related to TRAF 6.

5.2 Measures

Three independent variables (Information Filtering, Task Complexity, Cognitive Fit) were measured as dichotomous variables (0 or 1). For example, 0 was assigned when the information was not filtered through pruning and 1 was assigned when the information was filtered through pruning.

Multi-item measures for cognitive load (Paas 1992; Sweller and Chandler 1994) were used for this study. Cognitive load can be measured by subjective or physiological variables. This study assumes that people are able to introspect on their cognitive processes and report the amount of perceived cognitive effort. Self-ratings might be questionable (Paas et al. 2003). However prior research has demonstrated that people are quite capable of giving a numerical indicators of their perceived mental burden (Gopher and Braune 1984).

Performance was measured by both accuracy of the answers and time taken to complete the task. Start/end times were checked and recorded by subjects. Subjects were asked to report the difference between these two times.

A single-item dichotomous measure per each task was created as a manipulation check. All measurement scales were validated through a pilot test. All items were anchored on a seven-point likert scale ranging from “very little” (1) to “very much” (7). Appendix A shows the measures used in the study.

6. RESULTS

6.1 Manipulation Checks

Manipulation checks were employed to ensure that the subjects used the various treatments such as textual and graphical information. In both simple and complex tasks, subjects with knowledge about the biological processes used in the experimental task may use this knowledge instead of using the information provided. Therefore, manipulation checks were used to ensure that the subjects used the given information to perform their tasks. A total of 234 cases that passed the manipulation checks were retained for subsequent analysis.

6.2 Partial Least Squares Analysis

Partial Least Squares (PLS) analysis was used for measurement validation and for evaluating the hypothesized paths in the research model. PLS analysis was considered appropriate for this study because it places minimal demands on sample size and distributional assumptions (Chin 1998). PLS analysis is considered suitable for testing a theoretical model in its early stages. Since this study is an initial attempt at empirical examination on the impact of information filtering and cognitive fit on cognitive load in the context of ontology pruning, the use of PLS analysis is appropriate. PLS is considered robust at handling data with different scale types. All exogenous variables (Information filtering, task complexity, and cognitive fit) were included as dichotomous variables in our model. These categorical variables were coded as 0 or 1 in PLS analysis, whereas other variables were measured differently. Cognitive load was measured by using a 7-point likert scale, and performance was measured by an accuracy rate ranging from 0 to 1.

6.3 Convergent Validity

The convergent validity of a reflective construct in the research is tested by the examination of standardized loadings. If standardized loadings are higher than 0.707, they will meet the condition that the shared variance between each measurement item and its latent construct exceed the error variance. As seen in the Table 2, loadings of each of the three items for cognitive load were higher than 0.790. All three items were retained in the analysis. However, time taken to perform the task did not load well to performance (-0.458), whereas accuracy loaded well (0.952). In addition R^2 value for performance measured by only time taken to perform the task was 0.02, which was too low to make the model statistically meaningful. Therefore, time taken to perform the task was dropped as an item that measures performance.

Table 2. Item Loadings and Construct Measurement Properties

Construct	Item	Standardized Loading	Cronbach's Alpha	Composite Reliability	Average Variance Extracted
Cognitive Load	CL1 CL2 CL3	0.820 0.848 0.823	0.778	0.870	0.690

To test the internal consistency, Cronbach's alpha, composite reliability, and average variance extracted (AVE) of cognitive load were examined. Cronbach's alpha (0.778) and composite reliability (0.869) values are higher than 0.7, the norm for reliability (Bearden et al. 1993; Yi and Davis 2003). As another measure of construct validity, AVE measures the amount of variance that a latent construct captures from its indicators relative to the amount of variance from measurement error (Fornell and Larcker 1981). According to Chin (1998), AVE of higher than 0.5 means that 50 percent or more variance of the indicators is accounted for and acceptable for analysis. AVE for cognitive load in this study is 0.689. Thus, convergent validity is established according to the evaluation of Cronbach's alpha, composite reliability, and AVE.

6.4 Task Order Effect

Each participant finished two tasks, a simple one and a complex one. To remove task order effect and repeated measure effect, the order in which simple and complex tasks were presented was randomized. The effect of the task order was further examined. The main effect and interaction effects of task order were found to be not significant. The test results confirm that there is no task order effect in the study.

6.5 Structural Model

The structural model was assessed by examining path coefficients, their significance level, and the R^2 values. Path coefficients indicate the strengths of the relationships between two constructs. The R^2 values show the amount of variance explained by the independent constructs (Barclay et al. 1995; Chin and Gopal 1995). The final dependent construct, Performance, had an R^2 value of 0.172, which indicates that the research model accounts for 17.2% of the variance in the dependent variable when performance was measured by accuracy. It is also instructive to examine the R^2 values for the intermediate variable in the structural model. The R^2 value for "Cognitive Load" was 0.287. R^2 values are high enough to make interpretation of the path coefficients meaningful.

Path coefficients in the structural model were computed with the entire sample, and bootstrapping method with 500 resamples was computed to obtain the t-values corresponding to each path, as shown in Fig. 7. The acceptable t-values for one-tailed tests are 1.64 and 2.33

at the significance levels of 0.05 and 0.01. Information filtering had a negative impact on cognitive load (β = -0.274, $p<0.01$), and therefore H1 was supported. Task complexity had a moderating impact on the relationship between information filtering and cognitive load (β = 0.243, $p<0.05$), thus supporting H2. Cognitive fit had a negative impact on cognitive load (β = -0.215, $p<0.01$), and therefore H3 was supported. Its impact and significance level was lower than those of information filtering. Cognitive load had a negative effect on performance (β = -0.415, $p<0.01$), supporting H4.

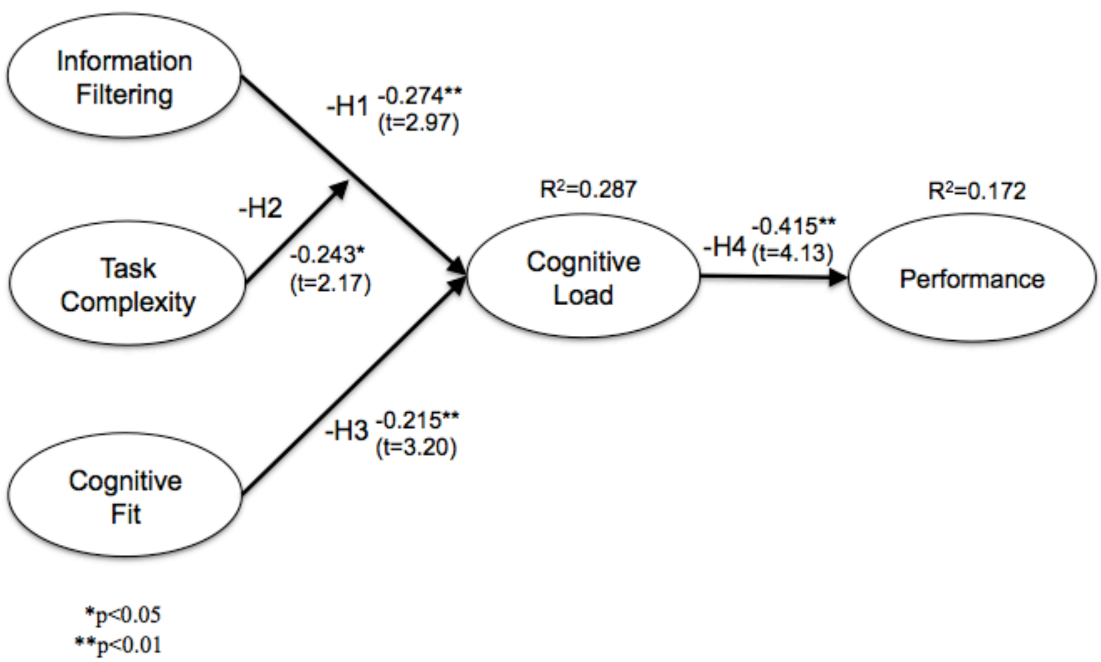


Figure 7: Structural Model

6.6 Discussion and Implications

This study empirically confirms that information filtering, task complexity, and cognitive fit can have significant effects on cognitive load, which affects task performance. Information filtering through pruning affects subjects' task performance measured by accuracy of answers. Compared with subjects with unfiltered information, subjects with filtered information perceive less cognitive load. The results of this study are consistent with the arguments by Malone et al. (1987) and Billinghurst and Starner (1999) and empirically confirm that information filtering using ontology pruning can reduce cognitive load. As biological and medical knowledge increases, users want to access information that is compact and relevant to their query. Therefore an ontology pruning feature needs to be integrated into the information system using a large bio-ontology such as GO and Open Biomedical Ontology.

Cognitive fit has a negative impact on cognitive load. The level of cognitive fit depends on how well given information represents a mental representation of a subject dealing with a task. The result is consistent with the findings by Vessey (1991) who examined the difference between the two different representations: table and text. Our study empirically confirms that this finding applies in the context of using ontology information. Understanding and analyzing information in bio-ontology requires mental mapping of structural information. Therefore, information systems that use bio-ontology should provide information in graphical format rather than textual format if the requested information is structural. If a bio-ontology based information system is used for other purposes such as information tagging, information in textual format would be appropriate.

The results show that effectiveness measured by the rate of accuracy is statistically significant whereas efficiency measured by the time taken to complete the task is not significant. People with significant cognitive load may not want to spend a lot of time on the given task. Instead of expending serious effort to solve a very complex problem, they may attempt to answer and solve it quickly. However, effectiveness of task performance can be increased when cognitive load is low. This suggests that information systems should help reduce cognitive load by filtering out unnecessary information and presenting information in an easy-to-understand format for a given task.

6.7 Limitations

Laboratory experimentation provides a highly controlled environment for hypothesis testing while it has a few methodological limitations. First, the experiment of this study is based on a scenario of using GeneOntology to find biological processes related to certain genes. There are several other uses of Gene Ontology that were not included in the study. Second, the use of student subjects may limit the generalizability of the results. However many prior studies on cognitive load establish the validity of using student subjects in similar settings (Moreno and Valdez 2005; Tuovinen and Sweller 1999). Third, this study measured a subject's subjectively measured cognitive load rather than objective cognitive load. Several examples of objective measurements include measures of heart activity, brain activity, and eye activity (Paas et al. 2003). However, objective measurement of cognitive load was not feasible in this study. Also, prior research has established the appropriateness of using subjectively-measured cognitive load (Gopher and Braune 1984). Fourth, task complexity has a positive impact on cognitive load, but its impact is relatively minor compared with the other two factors: information filtering and cognitive fit. Task complexity was treated as a dichotomous variable in this study. If a more complex task had been used, the impact to cognitive load might have been stronger. Finally using subjects with two different groups may generate a homogeneity issue. This study uses subjects with either IS or biology background. However, task performance of these two groups is not statistically significant.

7. CONCLUSIONS

This research addresses an information overload issue by introducing, implementing and evaluating an ontological approach. Ontology is considered a critical technology in the field of information systems (Fensel and Brodie 2003). Although ontology has a potential to make information systems intelligent, the use of large ontologies is severely restricted by the limitations of human information processing. Therefore, the ability to prune ontologies to retrieve only relevant concepts and present them in an appropriate format is essential for successful use of an ontology. Our research provides an approach and tools that address this need.

Although recent research efforts in bioinformatics have resulted in the development of several large bio-ontologies (GO, UMLS), it has become increasingly difficult to retrieve relevant knowledge from such large ontologies. Our research suggests that with smaller amount of relevant information in an appropriate format users can perform better.

The prototype developed in the study demonstrates the feasibility of using pruning methods to retrieve relevant knowledge from large ontologies. A pruning algorithm that significantly extends existing approaches has been developed. Developers of ontology tools like AmiGO can benefit from our research by incorporating our pruning method. This research also develops a useful tool that can be used by researchers in the field of bioinformatics to retrieve relevant knowledge from large, complex bio-ontologies. The pruning approach proposed in this study can be applied successfully to other large ontologies that contain instances, such as Cyc, UMLS or most OWL ontologies. In order to apply the pruning method to these ontologies, our approach has to be adapted to use the language used to represent the ontology (such as CycL for Cyc).

Our study extends the cognitive load theory through the introduction of three new constructs: information filtering, task complexity, and cognitive fit. All these three constructs are found to be important factors affecting cognitive load. In particular, information filtering among the three factors has the most significant impact on cognitive load. Although prior studies suggest that filtered information enhance task performance (Sabou et al. 2005) (Maedche and Volz 2001), it has not been empirically tested in the context of use of structured information such as ontologies. This study builds an ontology pruning prototype to empirically test how information filtering affects task performance by reducing cognitive load. Furthermore, this study empirically compared with other factors, which affect cognitive load.

This study is inter-disciplinary in nature, as it draws from varied fields such as information systems, computer science, psychology, and biology. The IS perspective helps identify significant problems of interest to both research and practice that may be addressed using ontology pruning. A pruning method drawn from computer science is extended to develop GOP that addresses the challenges in using ontologies in bioinformatics. Theories from cognitive science are adopted to develop a theoretical model and hypotheses to explain how information filtering, task complexity, and information format affect task performance through cognitive load. This study illustrates how an inter-disciplinary approach can be used in design science research.

APPENDIX

General Instructions: You will be asked to read a bio-ontology and fill out the survey questions. Bio-ontology is a well-organized biological information. It is very important to answer all of the questions included in the survey, without leaving out any questions.

The two tasks you are asked to do involves TRAF 6 and TBK1.TRAF 6 is known to get involved in three biological processes: 1) Positive regulation of I-kappaB kinase/NF-kappaB cascade, 2) Positive regulation of T cell cytokine production, and 3) T cell receptor signaling pathway.

The task you are asked to do requires two steps. First, check your starting time and examine the attached ontology document about biological processes. Second, fill out the questionnaire.

How to read ontology document

is_a

The *is_a* relationship is a simple class-subclass relationship, where A *is_a* B means that A is a subclass of B;

For example, **nuclear chromosome** *is_a* **chromosome**.

```
intracellular non-membrane-bound organelle (GO: 0043232)
  [i] chromosome (GO: 0005694)
       [i] nuclear chromosome (GO: 0000228)
```

Each group received different type of data for 1st and 2nd tasks.

1st Task

1. Enter the current time in 00:00 (AM or PM) format.
2. Write down the IDs (e.g. 0050852) of all upper biological processes of ‘positive regulation of T cell cytokine production’ and ‘T cell receptor signaling pathway’ .
3. Write down the IDs (e.g. 0050852) of all upper biological processes of ‘positive regulation of T cell cytokine production’, ‘T cell receptor signaling pathway’, and 'positive regulation of I-kappaB kinase/NF-kappaB cascade’.
4. How many biological processes exist under "is_a" relationship between ‘positive regulation of T cell cytokine production’ and ‘positive regulation of production of molecular mediator of immune response’ (exclude two processes mentioned in this question)?
5. How many biological processes exist under "is_a" relationship between ‘T cell receptor signaling pathway’ and ‘activation of immune response’ (exclude two processes mentioned in this question)?
6. Write down the IDs (e.g. 0050852) of the biological process that is the 1st upper class shared by ‘positive regulation of I-kappaB kinase/NF-kappaB cascade ‘ and ‘T cell receptor signaling pathway’.

7. Write down the ID (e.g. 0050852) of biological process which is the 1st upper level process shared by both ‘positive regulation of T cell cytokine production’ and ‘positive regulation of I-kappaB kinase/NF-kappaB cascade’.
8. Write down the ID (e.g. 0050852) of the 1st upper level process that is shared by both ‘positive regulation of immune response’ and ‘positive regulation of immune effector process’.
9. Write down the ID (e.g. 0050852) of the 2nd upper level process shared by both ‘positive regulation of leukocyte mediated immunity’ and ‘positive regulation of production of molecular mediator of immune response’.
10. Write down the ID (e.g. 0050852) of the 2nd upper level process shared by both ‘positive regulation of adaptive immune response’ and ‘activation of immune response’.
11. After completion of step 10, enter the current time in 00:00 (AM or PM) format.
12. How many minutes did you spend for this task?
13. Please answer the following questions. (7-point scale ranging from very little to very much was indicated to subjects at the online questionnaire)
    a. How much mental effort was required to figure out the relationships among biological processes and to answer the questions?
    b. How difficult was it for you to answer the questions?
    c. How burdensome was this task?
14. Did you use the data provided by the ontologies to answer questions 2 thorough 10?

2nd Task

This is a 2nd task involving TBK1. TBK 1 is known to get involved in two biological processes: 1) I-kappaB kinase/NF-kappaB cascade and 2) positive regulation of I-kappaB kinase/NF-kappaB cascade.

1. Enter the current time in 00:00 (AM or PM) format.
2. Write down the IDs (e.g. 0050852) of all biological processes which exist under "is_a" relationship between 'protein kinase cascade' and 'positive regulation of I-kappaB/NF-kappaB cascade’ (exclude two processes mentioned in this question).
3. Write down the IDs (e.g. 0050852) of all biological processes which exist under "is_a" relationship between 'protein kinase cascade' and 'positive regulation of I-kappaB/NF-kappaB cascade’ (exclude two processes mentioned in this question).
4. Write down the IDs (e.g. 0050852) of biological processes which exist under "is_a" relationship between ‘regulation of I-kappaB kinase/NF-kappaB cascade’ and ‘intracellular signaling cascade’ (exclude two processes mentioned in this question).
5. Write down the IDs (e.g. 0050852) of biological processes which exist under "is_a" relationship between ‘Ikappa kinase/NF-kappaB cascade’ and ‘Cell communication’ (exclude two processes mentioned in this question).
6. Write down the IDs (e.g. 0050852) of biological processes which exist under "is_a" relationship between ‘Cellular process’ and ‘Cell communication’ (exclude two processes mentioned in this question).
7. After completion of step 6, enter the current time in 00:00 (AM or PM) format.
8. How many minutes did you spend for this task?

9. Please answer the following questions. (7-point scale ranging from very little to very much was indicated to subjects at the online questionnaire)
    a. How much mental effort was required to figure out the relationships among biological processes and to answer the questions?
    b. How difficult was it for you to answer the questions?
    c. How burdensome was this task?
10. Did you use the data provided by the ontologies to answer questions 2 though 6 above?
11. Please answer the following:
    a. Age
    b. Gender
    c. Academic/Business experience in Biology

REFERENCES

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., and Eppig, J.T. "Gene Ontology: tool for the unification of biology," *Nature Genetics* (25) 2000, pp 25-29.

Baker, P., Goble, C., Bechhofer, S., Paton, N., Stevens, R., and Brass, A. "An ontology for bioinformatics applications," *Bioinformatics* (15:6) 1999, p 510.

Baker, P.G. "An ontology for bioinformatics applications," *Bioinformatics* (15:6) 1999, pp 510-520.

Barclay, D., Higgins, C., and Thompson, R. "The partial least square(PLS) approach to casual modeling: Personal computer adoption and use as an illustration," *Technology* (2:2) 1995, pp 285-309.

Bearden, W., Netemeyer, R., and Mobley, M. *Handbook of marketing scales* Sage Publ., 1993.

Bhatt, M., Flahive, A., Wouters, C., Rahayu, W., Taniar, D., and Dillon, T. "A distributed approach to sub-ontology extraction," *Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on* (1) 2004.

Billinghurst, M., and Starner, T. "Wearable devices: new ways to manage information," *Computer* (32:1) 1999, pp 57-64.

Blake, J., and Bult, C. "Beyond the data deluge: data integration and bio-ontologies," *Journal of Biomedical Informatics* (39:3) 2006, pp 314-320.

Borman, W., and Motowidlo, S. "Expanding the criterion domain to include elements of contextual performance," *Personnel selection in organizations* (71) 1993, p 98.

Borst, W.N. "Construction of Engineering Ontologies for Knowledge Sharing and Reuse," in: *Centre for Telematics and Information Technology*, University of Tweenty, 1997

Broadbent, D. *Perception and communication* Pergamon Press London, 1958.

Buitelaar, P., Cimiano, P., and Magnini, B. "Ontology Learning from Text: Methods, Evaluation and Applications," *Computational Linguistics* (32:4) 2006, pp 569-572.

Campbell, D. "Task complexity: A review and analysis," *The academy of management review* (13:1) 1988, pp 40-52.

Chin, W., and Gopal, A. "Adoption intention in GSS: relative importance of beliefs," *ACM SigMIS Database* (26:2-3) 1995, pp 42-64.

Chin, W.W. "The partial least squares approach to structural equation modeling," in: *Modern Methods for Business Research,* G.A.Marcoulides (ed.), Lawrence Erlbaum Associates, Mahwah, 1998, pp. 295-336.

Conesa, J., and Olivé, A. "Pruning Ontologies in the Development of Conceptual Schemas of Information Systems," in: *ER2004*, Shangai, China, 2004.

Dennis, A., and Carte, T. "Using geographical information systems for decision making: Extending cognitive fit theory to map-based presentations," *Information Systems Research* (9) 1998, pp 194-203.

Fensel, D., and Brodie, M. *Ontologies: a silver bullet for knowledge management and electronic commerce* Springer, 2003.

Foltz, P.W., and Dumais, S.T. "Personalized information delivery: an analysis of information filtering methods," *Communications of the ACM* (35:12) 1992, pp 51-60.

Fornell, C., and Larcker, D. "Evaluating structural equation models with unobservable variables and measurement error," *Journal of marketing research*) 1981, pp 39-50.

Good, B.M., Tranfield, E.M., Tan, P.C., Shehata, M., Singhera, G.K., Gosselink, J., Okon, E., and Wilkinson, M. "Fast, cheap and out of control: A zero curation model for ontology development," *Pacific Symposium on Biocomputing* (11) 2006, p 128?139.
Gopher, D., and Braune, R. "On the psychophysics of workload- Why bother with subjective measures?," *Human Factors* (26) 1984, pp 519-532.
Gruber, T.R. "Toward principles for the design of ontologies used for knowledge sharing," in: *Formal Ontology in Conceptual Analysis and Knowledge Representation,* N. Guarino and R. Poli (eds.), Kluwer Academic Publishers, Deventer, The Netherlands, 1993.
Holsapple, C., and Joshi, K. "A collaborative approach to ontology design," *Communications of the ACM* (45:2) 2002, p 47.
Johnson, E.J., and Payne, J.W. "Effort and accuracy in choice," *Management Science* (31) 1985, pp 395-414.
Jones, Q., Ravid, G., and Rafaeli, S. "Information overload and the message dynamics of online interaction spaces: A theoretical model and empirical exploration," *Information Systems Research* (15:2) 2004, pp 194-210.
Kirschner, P. "Cognitive load theory: implications of cognitive load theory on the design of learning," *Learning and instruction* (12:1) 2002a, pp 1-10.
Kirschner, P.A. "Cognitive load theory: implications of cognitive load theory on the design of learning," *Learning and Instruction* (12:1) 2002b, pp 1-10.
Lord, P.W., Stevens, R.D., Brass, A., and Goble, C.A. "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation," Oxford Univ Press, 2003, pp. 1275-1283
Maedche, A., and Staab, S. "Semi-automatic engineering of ontologies from text," Proceedings of the 12th Internal Conference on Software and Knowledge Engineering, Chicago, USA., 2000.
Maedche, A., and Volz, R. "The ontology extraction and maintenance framework text-to-onto," Proceedings of the ICDM '01 Workshop on Integrating Data Mining and Knowledge Management, California, USA, 2001.
Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A., and Cohen, M.D. "Intelligent information-sharing systems," *Communications of the ACM* (30:5) 1987, pp 390-402.
Mizoguchi, R., Vanwelkenhuysen, J., and Ikeda, M. "Task Ontology for reuse of problem solving knowledge," in: *Proceedings of Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KBKS'95)*, IOS Press, Amsterdam, 1995, pp. 46-57.
Moody, D.L. "Cognitive load effects on end user understanding of conceptual models: An experimental analysis," *Advances in Databases and Information Systems*) 2004, pp 129-143.
Moreno, R., and Valdez, A. "Cognitive load and learning effects of having students organize pictures and words in multimedia environments: The role of student interactivity and feedback," *Educational Technology Research and Development* (53:3) 2005, pp 35-45.
Musen, M. "Domain ontologies in software engineering: use of Protege with the EON architecture," *Methods of Information in Medicine-Methodik der Information in der Medizin* (37:4) 1998, pp 540-550.
Navigli, R. "Automatically Extending, Pruning and Trimming General Purpose Ontologies," in: *International Conference on Systems, Man and Cybernetics*, Tunisy, 2002.
Nickerson, J., and Zenger, T. "A knowledge-based theory of the firm: the problem-solving perspective," *Organization Science* (15:6) 2004, pp 617-632.
OMG "OMG Revised Submission, UML 2.0 OCL," 2003

Oudes, A., Roach, J., Walashek, L., Eichner, L., True, L., and Vessella, R. "Application of Affymetrix array and Massively Parallel Signature Sequencing for identification of genes involved in prostate cancer progression," *BMC Cancer* (5:86) 2005.
Paas, F. "Training strategies for attaining transfer of problem-solving skill in statistics: a cognitive-load approach," *Journal of educational psychology* (84:4) 1992, pp 429-434.
Paas, F., Tuovinen, J., Tabbers, H., and Van Gerven, P. "Cognitive load measurement as a means to advance cognitive load theory," *Educational Psychologist* (38:1) 2003, pp 63-71.
Peterson, B.J., Andersen, W.A., and Engel, J. "Knowledge Bus: Generating Application-focused Databases from Large Ontologies," in: *KRDB'98*, 1998, pp. 2.1-2.10.
Ramanujan, S., Scamell, R.W., and Shah, J.R. "An experimental investigation of the impact of individual, program, and organizational characteristics on software maintenance effort," *The Journal of Systems & Software* (54:2) 2000, pp 137-157.
Rowley, J. "Product search in e-shopping: a review and research propositions," *Journal of Consumer Marketing* (17:1) 2000, pp 20-35.
Ruiz, F., Vizcaino, A., Piattini, M., and GarcÌa, F. "An ontology for the management of software maintenance projects," *International Journal of Software Engineering and Knowledge Engineering* (14:3) 2004, pp 323-349.
Sabou, M., Wroe, C., Goble, C., and Mishne, G. "Learning domain ontologies for Web service descriptions: an experiment in bioinformatics," *Proceedings of the 14th international conference on World Wide Web*) 2005, pp 190-198.
Stevens, R., Horrocks, I., Goble, C., and Bechhofer, S. "Building a reason-able bioinformatics ontology using OIL," In Proceedings of the IJCAI-2001 Workshop on Ontologies and Information Sharing, 2001, pp. 81-90.
Swartout, W.R., Patil, R., Knight, K., and Russ, T. "Toward Distributed use of Large-Scale Ontologies," in: *Proc. Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop, Canada*, 1996.
Sweller, J. "Cognitive load during problem solving: Effects on learning," *Cognitive Science* (12) 1988, pp 257-285.
Sweller, J., and Chandler, P. "Why Some Material Is Difficult to Learn," *Cognition and Instruction* (12:3) 1994, pp 185-233.
Tuovinen, J., and Sweller, J. "A comparison of cognitive load associated with discovery learning and worked examples," *Journal of educational psychology* (91:2) 1999, pp 334-341.
Uschold, M., and Gruninger, M. "Ontologies: Principles, methods and applications," *The Knowledge Engineering Review* (11:02) 2009, pp 93-136.
van Merriëboer, J.J.G., and Sweller, J. "Cognitive Load Theory and Complex Learning: Recent Developments and Future Directions," *Educational Psychology Review* (17:2) 2005, pp 147-177.
Vessey, I. "Cognitive fit: A theory-based analysis of the graphs versus tables literature," *Decision Sciences* (22:2) 1991, pp 219-240.
Vessey, I., and Galletta, D. "Cognitive fit: An empirical study of information acquisition," *Information Systems Research* (2:1) 1991, pp 63-84.
Volz, R., Studer, R., Maedche, A., and Lauser, B. "Pruning-based identification of domain ontologies," *Journal of Universal Computer Science* (9:6) 2003, pp 520-529.
Wood, R. "Task complexity: Definition of the construct* 1," *Organizational behavior and human decision processes* (37:1) 1986, pp 60-82.

Yeh, I., Karp, P.D., Noy, N.F., and Altman, R.B. "Knowledge acquisition, consistency checking and concurrency control for Gene Ontology (GO)," Oxford Univ Press, 2003, pp. 241-248

Yi, M., and Davis, F. "Developing and validating an observational learning model of computer software training and skill acquisition," *Information Systems Research* (14:2) 2003, pp 146-169.

Zhang, W., Morris, Q., Chang, R., Shai, O., Bakowski, M., and Mitsakakis, N. "The functional landscape of mouse gene expression," *J Biol* (3:21) 2004.

Chapter 4

The use of ontology in knowledge intensive tasks:
Ontology Driven Retrieval of Use Cases

Abstract

Use cases are commonly used to represent customer requirements during systems development. In a large software development environment, finding relevant use cases from a library of past or related projects is a complex, error-prone and expensive task. This study proposes an ontological methodology to support use case retrieval in an interactive manner. The architecture of a prototype system that implements this methodology is presented. To evaluate the proposed approach, this study develops a research model and hypotheses based on interaction theory. These hypotheses are empirically tested using a laboratory experiment. Our study suggests that a system which interacts with a user intelligently reduces cognitive load and increases self-efficacy and satisfaction.

Keywords
Ontology, use case, perceived interaction, cognitive load, self-efficacy

1. INTRODUCTION

Requirements engineering (RE), which is concerned with acquiring and analyzing customer requirements, is a critical activity in software development. Use cases are gaining popularity in RE to represent customer requirements due to their simplicity and the use of natural language that facilitates the interaction between analysts and customers. In complex and large-scale systems development, the quantity of use cases that are needed to fully specify customer requirements can grow tremendously. Therefore, the ability to manage the development and reuse of use cases will significantly enhance project success. Specifically, appropriate techniques for the development, reuse, and modification of use cases can tremendously enhance productivity and project success.

The use of natural language in use cases, while arguably the primary reason for their popularity, also poses interesting challenges. Use cases expressed in natural language inherit some of the major problems with natural language based specifications; i.e. they are more likely to be inherently imprecise, ambiguous, incomplete and inconsistent. Therefore, approaches that accurately capture the meaning of use cases will significantly improve the ability to manage them. Our study uses an ontological approach to improving the retrieval of use cases. This approach can enhance the ability to retrieve use cases that are relevant to a current project from a repository developed in past and similar projects.

It is well established in the literature on software reuse that significant benefits in software development productivity can be gained by reusing artifacts developed early in the lifecycle rather than late in the lifecycle. We suggest that the reuse of use cases provides such an opportunity for significant savings. Therefore, we focus on improving the ability to retrieve use cases from a repository, which is a critical step in facilitating their reuse. Current Computer Aided Software Engineering (CASE) tools provide only keyword based search capability to retrieve use cases. In contrast, our research proposes an approach which draws from concepts used in the development of a Semantic Web.

The Semantic Web (Berners-Lee et al. 2001) refers to the Internet of the future. The goal of the Semantic Web is to provide information with a well-defined meaning for machine-to-machine as well as machine-to-human communication. The Semantic Web approach uses ontologies to achieve this goal. Ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where ontology refers to a systematic account of existence (Gruber 1993). By adding a well-defined meaning with ontological information, the

Semantic Web promises to provide better and meaningful search for information to both machines and humans.

The goal of our research is to apply the Semantic Web approach to improve the ability to retrieve relevant use cases in response to user queries. The creation and management of use cases expressed in natural language is often a difficult task in system development. We posit that ontological information will help overcome problems caused by the use of natural language in their specification and provide better search results to users. The primary research question addressed in this research is "How does the use of ontologies improve satisfaction in complex knowledge intensive organizational tasks such as use case retrieval IS development?"

To overcome the problems inherent in complex tasks such as the retrieval of use cases specified in natural language, prior research in RE has proposed several approaches. Sutton (2000) proposes mutual learning that results from frequent and close interaction through prototyping and rapid application development. Park et al (2000) argue that syntactic parsing may be used to analyze requirements more accurately. To improve requirement verification and validation, some researchers suggest the use of restricted formal expressions for requirements (Marcia and Pulman 1995; NE and U 2003). Also, recently researchers have used an ontological approach in requirement engineering to improve requirements analysis (Kaiya and Saeki 2005). They found that the ontological approach was useful in detecting incompleteness and inconsistency in requirements specification, measuring the quality of a specification and predicting requirements changes. While Kaiya and Saeki (2005) are concerned with the analysis of informal specifications, our approach uses an ontology to improve queries used to retrieve use cases, and is novel in the area of requirements engineering.

This paper is structured as follows. The next section discusses related research. The proposed ontological approach is introduced in section 3. Section 4 discusses the development of queries based on an ontology. The architecture for a system that implements the proposed approach is presented in section 5. The design of an experiment that evaluates the effectiveness of the proposed approach is introduced in section 6. The research model used in the experimental study is discussed in section 7. The results of experiment are presented in sections 8 and 9. Conclusions and future work are discussed in section 10.

2. RELATED RESEARCH

Ontology is defined as the explicit specification of a conceptualization (Gruber 1993), which is an abstract and simplified view of the world we want to represent. Obviously, due to the wide variations in "the world we want to represent", there exists a wide range of different kinds of ontologies (Guarino 1998; Lassila and McGuinness 2001; Poli 2002). A general ontology contains general information rather than specifics relevant to a particular context. General ontologies tend to contain information, such as time or space, that is independent of any domain, as well as general information dependent on a particular domain. Whereas general information is useful in common sense reasoning, domain information provides semantics that will be helpful in understanding that domain.

The creation of use cases is often the first step in the acquisition of requirements from users. Its role as an effective communication vehicle to capture requirements from users is a reason for its increasing popularity. Designers develop system design artifacts like state transition diagrams and class diagrams on the basis of use cases. Thus, use cases often represent a critical starting point in the development lifecycle. When stakeholders need to examine the relationships between the actual implementation and system requirements, they may rely on use cases that document requirements. Thus, from a RE perspective, a use case is a key artifact that is created and used throughout the processes of systems development.

A common task in RE is the search and exploration of requirements (which may be documented as use cases) that were created in earlier phases of a project or in other similar projects. These activities are supported only to a limited extent by current RE and CASE tools. These tools typically provide keyword based search capabilities similar to those used in web search engines like Yahoo or Google. Typically, such searches are not very helpful because of the ambiguities inherent in natural language. To overcome this problem, ontology based searching has been suggested by prior research. For example, Storey et al. (2008) proposed a context-aware query processing methodology called CONQUER which uses lexicons and ontologies to improve web query results. Researchers argue that real progress can be made in Software and Systems engineering when this approach is integrated with popular notations such as UML. Saeki (2004) introduces an ontology-based technique to support software requirements elicitation and to compose software from reusable architectures, frameworks, components and software packages. Their work focuses on semantic processing of requirements and reusable artifacts. Kaiya and Saeki (2005) use an ontology that consists of a thesaurus and a set of inference rules to detect incompleteness and inconsistency, measure the quality of a specification, and predict requirements changes. Although prior research has tried to address issues related to resolving ambiguities in requirements by applying linguistic techniques to use case analysis, limited attention has been paid to the use of domain knowledge in these activities (Fantechi et al. 2002; Fantechi et al. 2003). Our work addresses this research gap by using a domain ontology to support the retrieval and reuse of use cases.

3. THE USE OF AN ONTOLOGICAL APPROACH TO USE CASES QUERY

Our approach uses a combination of linguistic, semantic and extensional knowledge to improve the queries of use cases. We use ResearchCyc ontology in this research because it appears to be the only ontology that contains linguistic, semantic and extensional knowledge. ReserchCyc, which is a complete version of the Cyc knowledgebase (Cyc) for the scientific community, contains more than 2 million assertions (facts and rules) describing more than 250,000 terms and including nearly 15,000 predicates (Matuszek et al. 2006). The quantity and quality of the information it contains about actions are superior to those of other ontologies. For example, ResearchCyc contains a taxonomy of more than 6,000 actions. Since use cases often specify actions that are supported by a system, ResearchCyc is an excellent candidate for supporting the creation and use of use cases. Linguistic information (such as synonyms) is used to deal with some ambiguities of the natural language. Semantic information is used to develop intelligent queries, as can be seen in the following example: Suppose a designer is interested in use case diagrams that describe the rental of a GPS in a system for making reservations for a rental car. If the repository of specifications does not have a specific use case for renting a GPS, a use case that explains how to make a car reservation is likely to be of interest to the designer. Current ontological approaches that deal with requirements use a thesaurus to support the query process. Since a thesaurus does not contain semantic or extensional information, advanced inferences cannot be made with such approaches. On the other hand, semantic information present in general ontologies may provide a more powerful ability in the retrieval of use cases.

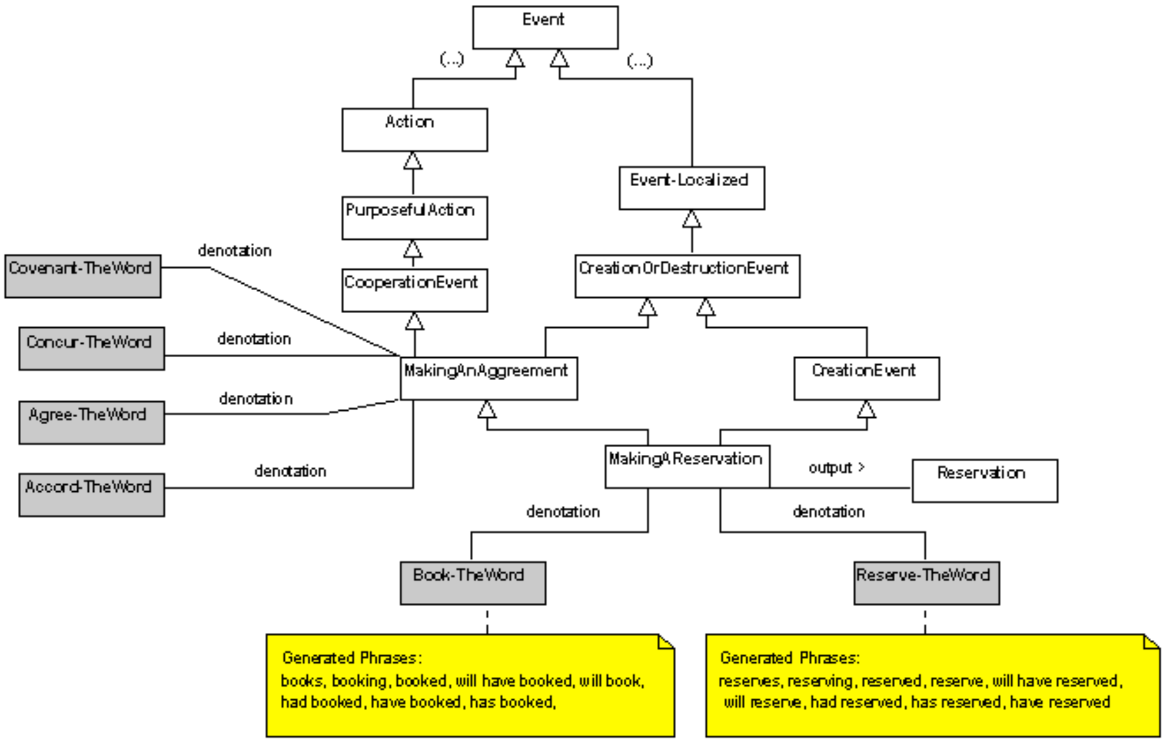


Figure 1: Fragment of ACTION ontology (the gray classes denote lexical information)

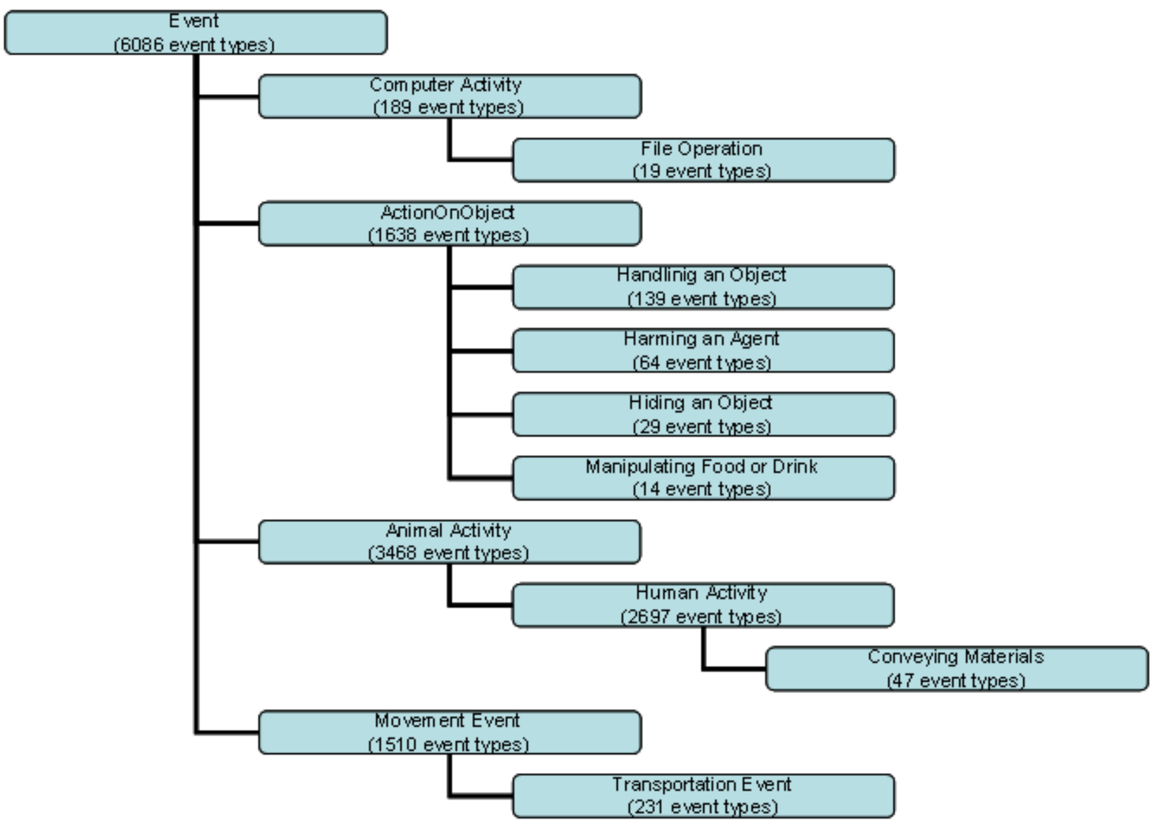


Figure 2: The main event types of ACTIONS and the number of event types they contain

3.1 ACTION: The ACTIons ONtology

ResearchCyc contains information about several domains. However, concepts that represent actions that use cases define are of interest in our research. Therefore, we create Actions Ontology (ACTION) from the ResearchCyc ontology by selecting all the sub types of a concept that represents events. In ResearchCyc, an event is defined as the dynamic situation in which the state of the world changes. ACTION ontology is created by including concepts that are related to events or any of its subtypes through relationship types. The rest of concepts contained in the ResearchCyc ontology have been deleted using a pruning algorithm (Conesa and Olivé 2006).The taxonomy of Events defined in the ontology contains more than six thousand different kinds of events that cover most of the actions that are commonly supported by information systems. In addition, ACTION also contains lexical information. In particular, it contains words that denote each of the actions. Figure 1 shows a fragment of the ACTION ontology. Concepts in grey represent the linguistic information related to the events in the ontology. For example, MakingAReservation event is related to the following words: book and reserve. The ontology also contains the different ways (conjugations) in which those words may be used in a text or a query. For example, the word book may be written as “books, booking, booked, will have booked…”. ACTION covers information on many domains (e.g. rental processes, terrorist actions, piracy actions etc.). Figure 2 shows some of the most important topics of ACTION and the number of subtypes associated with each topic.

3.2 The Domain Ontology

Even though ACTION can be used to retrieve use cases, sometimes it may not provide sufficient information to generate a satisfactory answer to a users’ query. For example, suppose the designer searches a use case repository with a query: How do I rent a GPS? As mentioned earlier, the system can infer that the use case “Make a Reservation” is relevant for renting a GPS. To perform this inference, the system needs to know that in the context of this query, the GPS is a part of a car. Since ResearchCyc does not contain the information that a GPS may be part of a Car, the use of a domain ontology focused on automobiles is necessary. Thus, the use of a domain ontology (or ontologies) relevant to the domains that the use cases deal with will be very useful in improving the inferences that are necessary for effective RE.

3.3 WordNet

WordNet is being maintained as a semantic lexicon for the English language. It groups English words into sets of synonyms with short, general definitions. In addition, it records the various semantic relations between these synonym sets. The purpose of WordNet is twofold. One is to produce a combination of dictionary and thesaurus that is more intuitively usable. The other is to support automatic text analysis and artificial intelligence applications. In our study, WordNet can be used to provide lexical information, which may not be available in ACTION and domain ontology.

4. ONTOLOGY DRIVEN REQUIREMENTS QUERY

The proposed methodology uses semantic and linguistic knowledge to identify the use cases that fit the query of the user. Therefore, any web query methodology that deals with linguistic and semantic knowledge can be adapted for this study. The adaptation only requires replacing the web query engine with a use case query engine. In the following sections, we describe a query methodology and demonstrate the improvements that can be achieved with the use of ontologies. The methodology is similar in spirit to the approach used by a web query methodology presented in (Conesa et al. 2006). The methodology is distinctive in that the interaction between user and the system is integral process in refining original queries.

The proposed query methodology is composed of four phases:

4.1 Query Parsing Phase

This phase receives a query expressed in natural language by the user. The nouns, noun phrases and verbs are identified from the initial query using a POS Tagger (Mason Accessed on Jan 7, 2005). The output of this phase is a set of query terms $w_1, ..., w_n$ that are used as the initial query.

4.2 Concepts identification Phase

The input of this phase is a set of words related to the use cases that are of interest to the user. The purpose of this phase is to find the concepts of ACTION and domain ontologies that represent the words of the query, or are closely related to them.

We say that a concept C represents a query term w_i if:

1. w_i is part of the noun of the concept, or
2. There is any linguistic relationship between the word w_i and the concept C in the ontology (the relationship types of ACTION that define linguistic information are *termStrings*, *denotation* and *genStringAssertion*), or
3. a WordNet concept which is synonym of w_i satisfy any of the two previous conditions.

The output of this phase is a set of the concepts that represent the candidate query words of the input $\{C_1, ..., C_m\}$.

4.3 Interaction Phase

The interface module receives the output of the concepts identification phase and presents it the user. The purpose of this phase is to allow the user to select relevant concepts in order to refine his/her query. The interface module sends the refined query to the inference module.

4.4 Query Inferences Phase

The input of this phase is a set of concepts provided by the interaction phase. The purpose of this phase is to select all the use cases relevant to the input concepts.

Suppose, in the context of the rental car information system, the user wants to find use cases that deal with the reservation of the 'child seats' and uses the query term "children car seat rental". The system looks up the concept in the ontology library. It identifies that 'reservation', 'automobile', and 'seat' are the relevant concepts. The user selects the concepts that may help refine the original query. Then, the system searches a library of use case and sends the search results to the interface module which presents the information to the user.

5. PROTOTYPE

5.1 Architecture

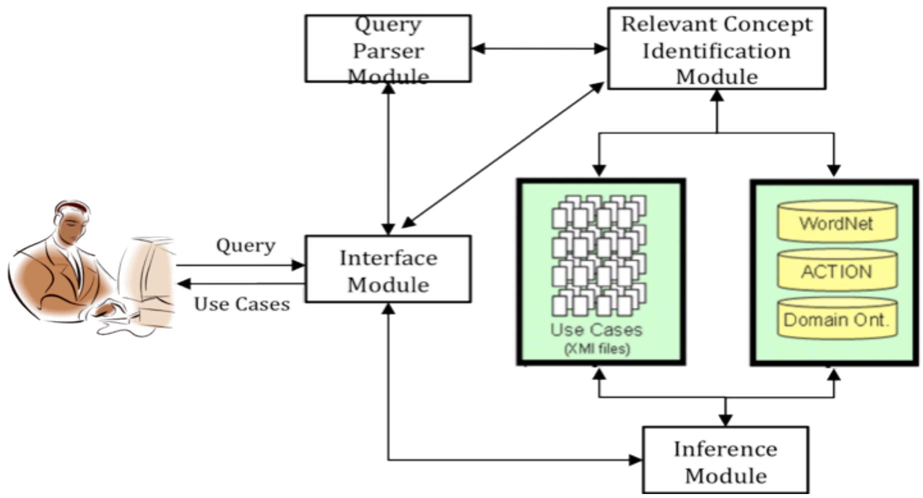


Figure 3: System Architecture

Figure 3 shows the architecture of our prototype system for ontology-driven use case retrieval. It consists of: a) Query Parser Module, b) Relevant Concepts Identification Module, c) Inference Module, and d) Interface Module. The Interface Module enables the interaction between users and the system. The Query Parser Module captures the user's query and parses it to return the part-of-speech for each term. The Relevant Concepts Identification Module interacts with ACTION, domain ontology (if necessary) and WordNet. For each query, it obtains related concepts from ACTION and domain ontologies. To obtain this information, linguistic relationships such as synonyms are used. The output of this module is sent to the Interface Module. The Inference Module receives the selected concepts and finds relevant use cases. Finally, the Interface Module will present selected use cases to the user.

5.2 Domain Ontology

To illustrate the proposed approach, a domain ontology in the domain of job search was created. This ontology was created in the OWL Language using the Protégé tool (Noy et al. 2001).. Figure 4 shows a simplified conceptual representation of the ontology.

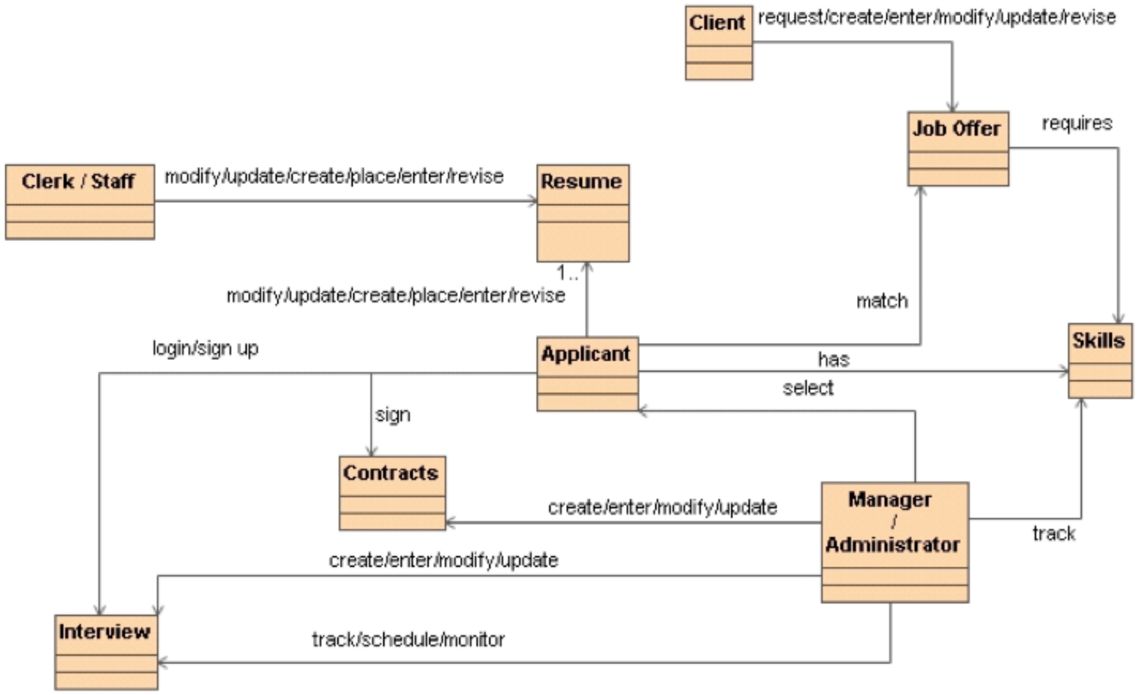


Figure 4: Conceptual Representation of Domain Ontology

The domain ontology shown at figure 4 identifies relationships among key terms and synonyms. These relationships are used to identify terms that are semantically related terms used in a user query. Multiple terms in a class or a property indicate that the terms have synonyms. When a user's key term has related terms and synonyms, these are displayed to the user (see Figure 5).

5.3 Implementation

The prototype is implemented as a web application using JSP (Java Server Pages). This development environment was chosen because it would make the system portable and easily accessible through the World Wide Web. The web application interacted with the MySql database of Use Cases and the domain ontology in OWL web ontology language (W3C 2004).

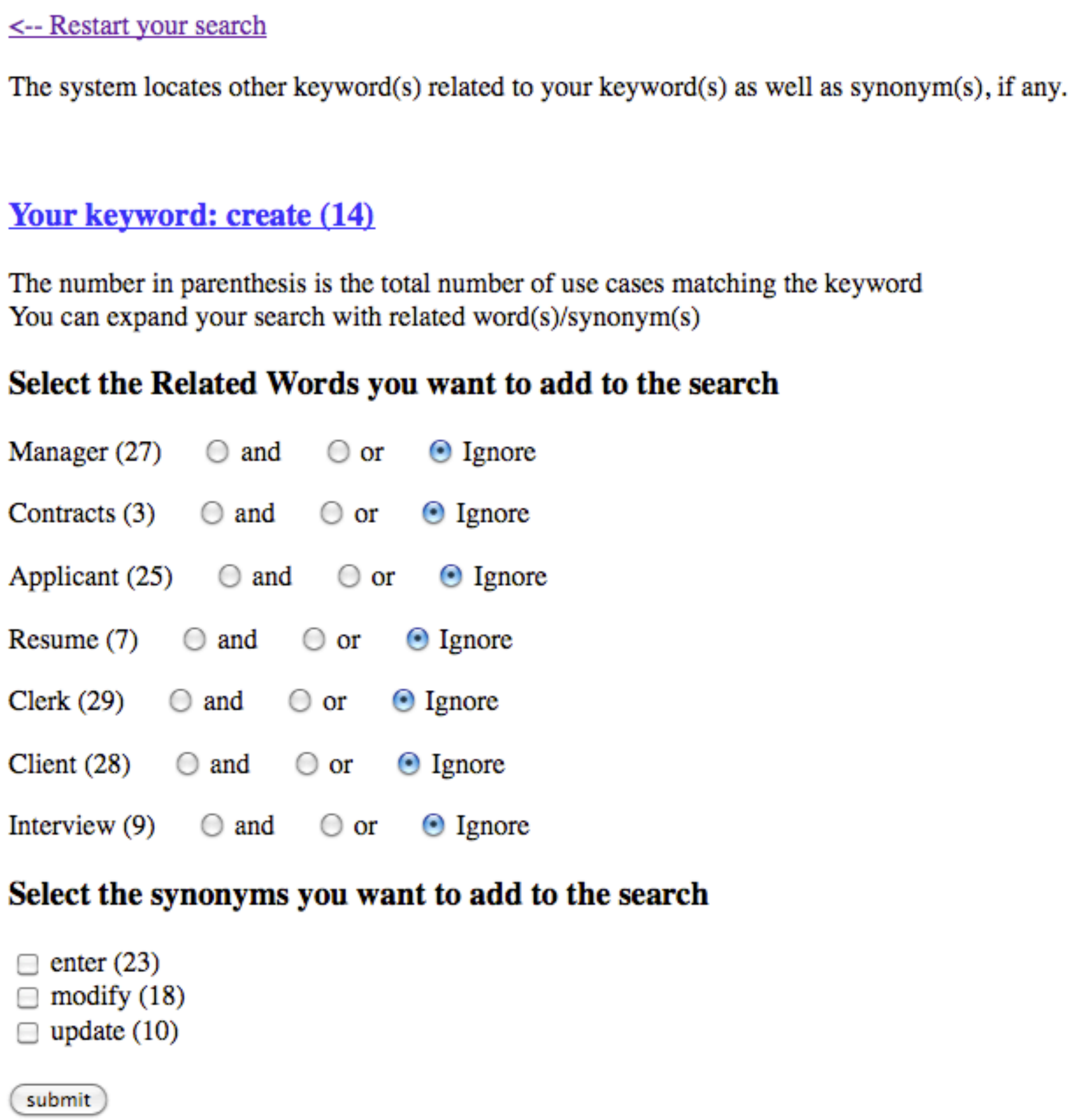


Figure 5: Ontology Supported Interactive Prototype System

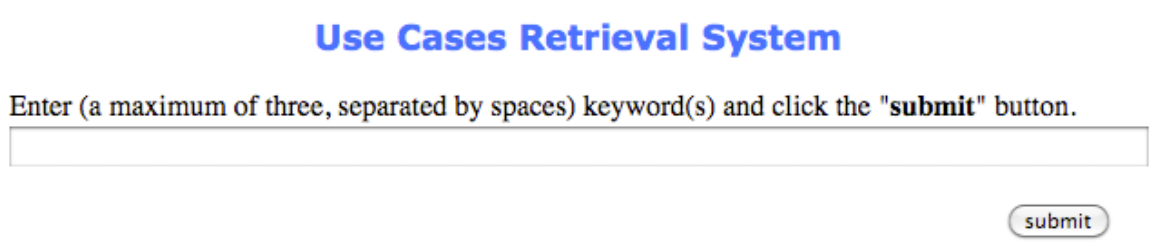


Figure 6: User Interface of the Ontology Supported Interactive Prototype System

On the client side, web pages were used to gather information from a user, such as an initial query keyword and relevant keywords. On the server side, several java servlet modules are used to parse multi keywords query and identify both the elements of the ontology and the synonyms related to the initial query. For the interaction with the domain ontology, the prototype uses the OWL API (http://owlapi.sourceforge.net/index.html), which is an open source Java tool that is used to read the domain ontology in OWL. The interaction with the ontology is shown at Figure 5.

6. EMPIRICAL EVALUTION OF THE EFFECTIVNESS OF THE PROPOSED APPROACH

The effectiveness of the proposed approach is evaluated using an experimental study. A research model for this study is shown in Figure 7. In this experiment, information filtering and perceived interaction are used as exogenous constructs whereas cognitive load, cognitive control, and satisfaction are identified as endogenous constructs.

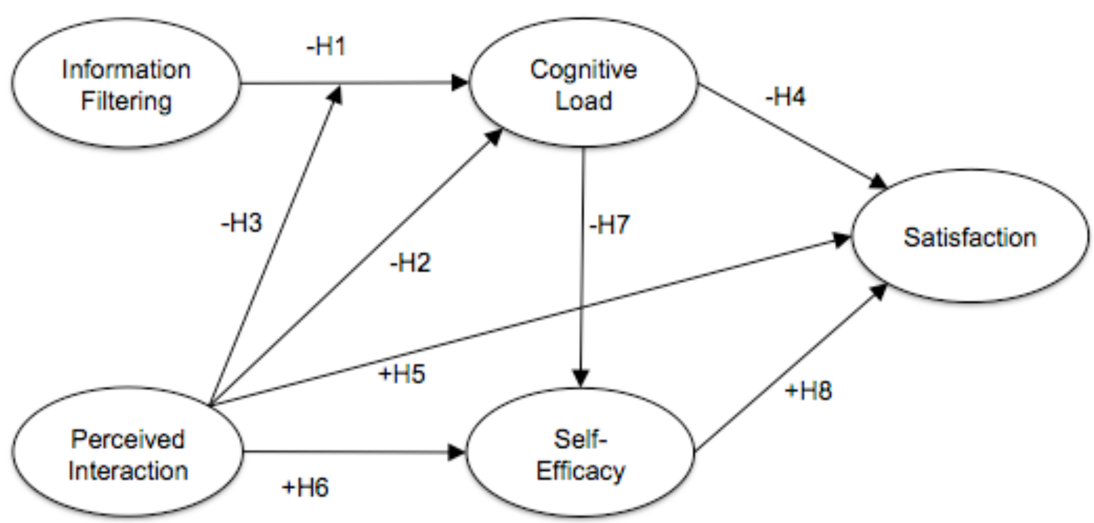


Figure 7: Research Model

Information filtering is a method for the delivery of relevant information (Malone et al. 1987). Relevant information can be retrieved after irrelevant information is filtered out. When users are provided the ability to receive only information relevant to a given context and discard irrelevant information the cognitive load involved in information processing may be reduced. The processing of unfiltered information increases the cognitive load faced by users. Information filtering reduces cognitive load by selecting only relevant parts from a larger set of information (Malone et al. 1987). Billinghurst et al (1999) note that information systems can be used to filter out unnecessary information thereby reducing cognitive load. Ontology can play an important role in information filtering in tasks such as information retrieval. Users may perceive a high cognitive load while identifying relevant search terms for use in information retrieval tasks. Often, users need to identify several keywords and select among them for effective information retrieval. This keyword generation and selection processes involve high cognitive load.

Ontology has been recognized as an effective tool in improving information retrieval (Klischewski 2006). The use of ontology helps overcome the limitations of keyword-based search by providing the ability to represent class hierarchies and relationships. The additional representational power provided by ontology can significantly improve the identification of keywords by expanding the queries with related, relevant terms. Prior literature on cognitive load and information filtering suggests that Information filtering will reduce cognitive load. Thus, we hypothesize

Hypothesis 1: Information filtering will negatively affect cognitive load.

Interaction is an action that occurs as two or more objects have an effect upon one another. In human-computer interaction, the interaction between users and computers occurs at the user interface which may be implemented in software or hardware. Interaction in information retrieval systems occurs when users communicate with computers by specifying queries, receiving results, and revising queries (say, by experimenting with different keywords). Prior research establishes that poorly designed human-computer interfaces can lead to unexpected problems such as misinterpretation of information. Therefore, much of the research in this area has focused on the design of better human-computer interfaces. However, recent studies note that while research that addresses procedural or functional aspects of interaction is important, more research is needed to address perceptional aspects of interaction. McMillan and Hwang (2002) identify three dimensions of perceived interaction: conversation, delay, and engaging. Our study subscribes to this new perspective on interaction. Perceived interaction rather than interaction as a feature of the system is the focus of this study.

Perceived interaction can affect cognitive load perceived by uses in information retrieval tasks. The high cognitive load associated with the generation and refinement of keywords in information retrieval can be reduced by the ability to perform these tasks interactively. In the absence of interaction, users do not receive any feedback on the appropriateness of the keywords they use for information retrieval. When the users can interactively perform these tasks, they may be able to successively refine their keywords and successfully complete their information retrieval tasks. Thus,

Hypothesis 2: Perceived interaction will negatively affect cognitive load.

The quality of the perceived interaction in information retrieval tasks can be significantly improved with the use of ontology. With the use of relevant knowledge provided by an ontology, users may quickly identify and refine keywords used, thereby reducing the cognitive load involved in these tasks. While the development of algorithms and performance measures to improve information retrieval have received much attention in prior research, the use of ontology to improve the quality of results in information retrieval tasks hasn't been examined adequately. Recognizing this gap, Storey et al. (2008) propose CONQUER, a methodology for context-aware query development. This research establishes the use of interaction and ontology in facilitating information retrieval. Through the interaction between the user and an ontology-supported retrieval system, users can improve their search queries. In similar spirit, Vallet et al (2005) adapt a vector-based ranking model, which takes advantage of an ontology to help users interactively improve their queries. These studies suggest that interaction moderates the relationship between information filtering and cognitive load.

These studies suggest that higher levels of perceived interaction accelerate the reduction of cognitive load when coupled with higher levels of information filtering. Thus,

Hypothesis 3: Perceived interaction moderates the effects of information filtering on cognitive load.

Cognitive load refers to the load on the working memory during problem solving, thinking and reasoning. It may be classified as: intrinsic cognitive load, extraneous cognitive load, and germane cognitive load (Sweller 1988; Sweller and Chandler 1994). Intrinsic cognitive load is determined by the interaction between the nature of material to be learned and the learner's expertise. Extraneous cognitive load is the extra load beyond the intrinsic cognitive load resulting from poorly designed instruction, whereas germane cognitive load is the load related to processes that contribute to the construction and automation of schemas (Paas et al. 2003).

Satisfaction is defined as ‘a judgment that a service provided a pleasurable level of consumption-related fulfillment’ (Oliver 1996). Thus, satisfaction is the user’s sense that consumption provides outcomes against a standard of pleasure versus displeasure. Anderson and Sullivan (1993) postulate that satisfaction can be “broadly characterized as a post-purchase evaluation of product quality given pre-purchase expectations.” Past research suggests that satisfaction is influenced by perceived performance of a product or a service (Cronin and Taylor 1994). Therefore, perceived quality and satisfaction need to be separated because these are different at a conceptual level (Kettinger and Lee 1994). In this research, satisfaction is considered stemming from transaction experiences (Parasuraman et al. 1994).

Prior studies on information retrieval suggest that a trade-off relationship between cognitive load and user satisfaction exists (Branting 2001; McSherry 2004). A study on consumer behavior finds that cognitive load decreases the tendency to choose the better quality option thereby reducing user satisfaction (Drolet and Frances Luce 2004). Back and Oppenheim (2001) assume that low cognitive load from a user-friendly interface of an information retrieval system can result in high user satisfaction.

Based on prior literature on cognitive load and information satisfaction, this study predicts that cognitive load will reduce satisfaction. Thus,

Hypothesis 4: Cognitive load will negatively affect satisfaction.

Prior research suggests that user interaction with the system has an impact on a participant’s satisfaction (Lamport 1993). Driver (2002) argues that interaction stimulated by online discussions may effectively enhance students' class experience and increase their satisfaction. Wells et al. (1999) note that information technology can support one-to-one customer interaction thereby increasing customer satisfaction.

Cognitive feedback theory provides a theoretical explanation of the relationship between perceived interaction and satisfaction. Sengupta and Te'eni (1993) define cognitive feedback as information about the decision maker's decision strategy and the extent to which the strategy is applied accurately. While outcome feedback describes the accuracy of a decision, cognitive feedback provides decision makers with insight into their decision processes (Balzer et al. 1989). Interactions between users and the system with the support of ontological knowledge can provide cognitive feedback, which would be increase user satisfaction. Thus,

Hypothesis 5: Perceived interaction will positively affect satisfaction.

Self-efficacy refers to one's belief in one's capability to perform a specific task (Bandura and Adams 1977). Social cognitive theory posits that neither inner forces nor external stimuli drive people to exhibit a certain behavior. The theory explains that human behavior, cognitive and personal factors, and environmental events all operate interactively with one another. Self-efficacy, defined as a person's judgment of his/her capabilities to perform a given task, is a key regulatory mechanism in this relationship. Bandura (1982) postulates that self-efficacy helps determine what actions to take, how much effort to invest, how long to persevere and what strategies to use in challenging situations. Prior studies support this proposition in a variety of settings such as technology acceptance (Agarwal et al. 2000; Venkatesh 2000), computer skill acquisition (Mitchell et al. 1994), and complex decision making (Wood and Bandura 1989).

Cognitive feedback theory suggests that feedback supported by information systems increases decision quality and confidence in individual decision making (Hogarth 1996) . When the user provides general and ambiguous terms during information retrieval system, the system may generate irrelevant results. In contrast, with an interactive system, user can refine their queries based on the feedback provided by intermediate results. Thus, perceived interaction increases the user's confidence in their capability or their self-efficacy. Thus,

Hypothesis 6: Perceived interaction will positively affect self-efficacy.

When people encounter difficult tasks, which require a significant effort and time to complete, they might lose their belief in their abilities to cope with those tasks. Prior research on e-learning environments identifies several variables as motivators of students. These include perceived importance, usefulness, and the value of engaging in a task (Pintrich and Schrauben 1992). When learners perceive the effort as a waste of energy or as unnecessary, they are not motivated to exert sufficient mental effort. Another important variable affecting a person's motivation to take challenging tasks is his/her preconceptions about the effort required to accomplish a task. Keller and Suzuki (2004) note that self-efficacy is an important component of motivation. Therefore, preconceptions on the effort required to complete a task affect not only motivation but also some characteristics of the learner such as self-efficacy.

Perceived cognitive load (e.g. perceived difficulty and complexity) about a task can affect people's self-efficacy. When people perceive a given task as very difficult and complex, they assume that the task requires a lot of effort and time. Then, people question whether they have the ability to invest such effort and time required for the task.

Problem solving requires cognitive effort and places a certain amount of load on working memory processes (Wood et al. 2000). For example, people who are good at puzzles or mathematics are likely to have high self-efficacy. They can handle tasks which require high cognitive load. This suggests that people with high self-efficacy can handle tasks requiring objectively high cognitive load. When people subjectively perceive low cognitive load from tasks, they believe they have the ability to successfully accomplish the tasks. Thus,

Hypothesis 7: Cognitive load will negatively affect self-efficacy.

Computer self-efficacy (CSE) is a two-level construct which operates at the general computing level (general CSE) and at the specific application level (application specific CSE). General CSE refers to an individual judgment of efficacy across multiple computer domains whereas application specific CSE refers to an individual perception of efficacy in using a specific application or system within the domain of general computing. Prior research on technology acceptance focuses on the effects of general CSE on users' attitude to a system (Venkatesh and Davis 1996)**.** Agarwal et al (2000) proposed a model which differentiated a general CSE and an application specific CSE. They empirically established that an application specific CSE has statistically more significant effect on users' attitude toward a system use and adoption. Our study subscribes to this view.

User satisfaction is considered as one of the most important measures of information systems success (Delone and McLean 1992). In particular, users with high-level of a general CSE perceive high satisfaction on the use of an information system. Prior research suggests that self-efficacy has a positive relationship with user satisfaction

(Henry and Stone 1994). Based on prior research on self-efficacy and satisfaction, we hypothesize

Hypothesis 8: Self-efficacy is positively associated with satisfaction.

7. RESEARCH METHOD AND DESIGN

7.1 Treatment, task, and prototypes

A laboratory experiment was conducted to test the causal relationships between the constructs in the research model. The experiment involves a two-factor, four-cell design with two exogenous variables: information filtering and interaction. Both information filtering and interaction were manipulated at two levels. However, perceived interaction was measured by multiple items developed by McMillan and Hwang (2002). All subjects in four cells were asked to find two relevant use cases in a given context. After the study instruments were developed, pilot tests were conducted to refine the treatments and validate the measures. Students taking courses in information systems at two large southeastern and northeastern universities in U.S. served as subjects. A total of 121 subjects participated this experiment of which 99 passed manipulation checks. Their mean experience with system analysis was 16 months. Sixty four percent of the subjects were male, and thirty six percent were female.

Subjects were randomly assigned to one of the four cells on the basis of the last digit of their birthday. The experimental task involved the retrieval of two use cases from a library of use cases.

Four prototypes were developed to provide two levels each of information filtering and interaction as shown at Table 1. The *first* prototype provided no interaction and no information filtering. Subjects used keywords to retrieve use cases relevant to their given task. To minimize the interaction between subjects and the system, subjects were allowed to use the retrieval system only one time. In addition, twenty seconds delay was added before the system delivers the result. The *second* prototype provided high levels of interaction and no information filtering. In this prototype, the subjects could return to home page and enter additional keywords. The *third* prototype supported no interaction, but provided information filtering with the use of an ontology. When a user enters a keyword, the retrieval system identified other related keywords and showed those keywords and matching use cases. In addition, twenty seconds delay was added before the system delivers the result. Finally the *fourth* prototype supported high levels of both interaction and information filtering. When users enter a keyword, the system suggests additional terms and synonyms. Subjects can choose to include these terms with the use of conjunctions or disjunctions (see Appendix A for screenshots of the four prototypes).

After finishing the task of finding relevant two use cases, subjects were asked to complete a questionnaire. The questions consisted of items that measured cognitive load, satisfaction, perceived interaction, and self-efficacy with the system.

Table 1: Group Design and Four Prototypes

		Information Filtering (n = number of participants)	
		No	Yes
Interaction	No	Prototype I (n=26)	Prototype III (n=18)
	Yes	Prototype II (n=23)	Prototype IV (n=32)

7.2 Measures

Information filtering was measured as a dichotomous variable (0 or 1). Perceived interaction was measured by multi-item measures along three dimensions of perceived interaction (i.e. conversation, delay, and engagement) (McMillan and Hwang 2002).

Multi-item measures for cognitive load were used for this study (Paas 1992; Sweller and Chandler 1994). Cognitive load can be measured as a subjective variable. This study assumes that people are able to introspect their cognitive processes and report the amount of perceived cognitive load. Prior research has demonstrated that people are quite capable of assessing their perceived mental burden involved in performing a task (Gopher and Braune 1984).

Multi-item measures for satisfaction were used for this study. Wixom and Todd (2005) propose measurements for information satisfaction and system satisfaction. This study uses measurement items for system satisfactions. Finally self-efficacy was measured by three items adapted from prior research (Johnson and Marakas 2000; Yi and Hwang 2003).

All measurement scales were validated through a pilot test. Items for perceived interaction were anchored on a seven-point likert scale ranging from "not at all descriptive" (1) to "very descriptive" (7). Items for cognitive load were anchored on a seven-point likert scale ranging from "very little" (1) to "very much" (7). Items for self-efficacy were anchored on a eleven-point likert scale ranging from "completely disagree" (0) to "completely agree"(10). Items for satisfaction were anchored on a seven-point likert scale ranging from "strong disagree" (1) to "strongly agree" (7). Based on the results of the pilot study, minor modifications were made to the survey design. The final survey included 22 items representing the four constructs identified in Figure 7. Appendix B shows the measures used in the study.

8. RESULTS

8.1 Manipulation Checks

Manipulation checks were employed to ensure that the subjects received the intended treatments for information filtering and interaction. Subjects were asked to report whether they used system that filtered information in completing their task. Subjects who received interaction treatment were asked whether they perceived the system to be interactive. A total of 99 cases that passed the manipulation checks were retained for subsequent analysis.

8.2 Partial Least Squares Analysis

Partial Least Squares (PLS) – a second-generation structural equation modeling technique – was used to evaluate the adequacy of the measurement model and then to test the hypothesized structural model. Three considerations motivated the choice of PLS.

First, it has minimal demands on sample size and distributional assumptions (Chin 1998). Regression may yield unstable results when the sample size increases standard error of their estimated coefficients.

Second, PLS' ability to handle items with different scales is superior to multiple regression and traditional path-analytic techniques. PLS is considered robust at handling data with different scale types. This study uses measurement items with different scales Information filtering was included as dichotomous in the model. This categorical variable was coded as 0 or 1 in PLS analysis whereas other variables were measured differently.

Perceived interaction, cognitive load, and satisfaction were measured by likert scales ranging from 1 to 7. Self-efficacy was measured by a likert scale ranging from 0 to 10.

Third, PLS analysis is considered suitable for testing a theoretical model in its early stages. Since this study is an initial attempt at empirical examination of the impact of information filtering and interaction on satisfaction through cognitive load and self-efficacy in the context of ontology supported information filtering, the use of PLS analysis is appropriate.

SmartPLS version 2.0.M3 was used for the analysis, and the bootstrap resampling method (with 500 resamples) was used to determine the significance of the paths within the structural model.

8.3 Measurement Model Assessment

Our research model has both reflective and formative constructs. This affected the manner in which convergent validity was assessed as we made no assumption that formative indicators will covary. Therefore, traditional methods for assessing construct reliability cannot be applied to formative constructs. Multicollinearity was examined for a formative construct (interaction). A variance inflation factor (VIF) value of interaction was calculated. Prior research recommends that VIF values for formative measures should be less than 10. Since all VIF values (shown in Table 2) are less than 10, there is minimal risk of multicollinearity and all items for perceived interaction were retained to preserve content validity.

We assessed whether the scales exhibit sufficient convergent and discriminant validity. Standardized loadings were examined to test convergent validity of constructs used in this research. Standardized loadings should be higher than 0.707 to meet the condition that the shared variance between each measurement item and its latent construct exceed the error variance. As seen in Table 3, loadings of items for all constructs were higher than 0.857. Therefore all items were retained in the analysis.

Table 2: Variance Inflation Factor for Formative Construct

Construct	Items	Variance Inflation Factor (VIF)
Perceived Interaction	CONV1	2.40
	CONV2	2.89
	CONV3	3.48
	CONV4	1.64
	DELAY1	4.10
	DELAY2	1.94
	DELAY3	4.50
	ENG1	3.78
	ENG2	2.63
	ENG3	2.28
	ENG4	2.14
	ENG5	1.64
	ENG6	2.83
	ENG7	2.83

To test the internal consistency, Cronbach's alpha, composite reliability, and average variance extracted (AVE) of cognitive load were examined. Cronbach's alpha and composite reliability values are higher than 0.7, which is a recommended minimum value for reliability (Bearden et al. 1993; Yi and Davis 2003). As another measure of construct validity, AVE measures the amount of variance that a latent construct captures from its indicators relative to the amount of variance from measurement error (Fornell and Larcker 1981). According to Chin (1998), AVE of higher than 0.5 means that 50 percent or more variance of the indicators is accounted for and acceptable for analysis. AVE for all constructs in this study is higher than 0.769. Thus, convergent validity is established according to the evaluation of Cronbach's alpha, composite reliability, and AVE.

Table 3. Item Loadings and Construct Measurement Properties

Construct	Item	Standardized Loading	Cronbach's Alpha	Composite Reliability	Average Variance Extracted
Cognitive Load	CL1 CL2 CL3	0.893 0.949 0.917	0.909	0.943	0.847
Self-efficacy	SE1 SE2 SE3	0.871 0.902 0.857	0.850	0.909	0.769
Satisfaction	SA1 SA2	0.981 0.980	0.961	0.981	0.962

To test discriminant validity, we conducted two tests. First, we compared AVE for each construct with the shared variance between all possible pairs of constructs (Fornell and Larcker 1981). AVE for each construct is higher than the squared correlation between the construct pairs. This means that more variance is shared between the latent construct and its block of indicators than with another construct representing a different block of indicators. Therefore discriminant validity is established as shown in Table 4.

Table 4. AVE versus Squares of Correlations between Constructs

Construct	Average Variance Extracted (AVE)	CL	SE	SAT
CL	0.847	-		
SE	0.769	0.253	-	
SAT	0.962	0.232	0.386	-

Second, we calculated each indicator's loading on its own construct and its cross-loading on all other constructs were calculated (Chin 1998). Three sub-constructs of interaction (i.e. conversation, no delay, and engagement) were added. The loadings for the intended indicators for each construct are higher than the cross-loadings for indicators from other constructs as shown in Table 5. Two items for interaction (i.e. CONV4 and ENG5) were dropped from further analysis because they did not load high. Each indicator has a higher loading with its intended construct than a cross-loading with any other construct.

Table 5 Items to Own Construct Correlation versus Correlations with Other Constructs

Construct		Item	1	2	3	4	5	6
Cognitive Load		CL1	**0.893**	-0.440	-0.259	-0.501	-0.432	-0.420
		CL2	**0.950**	-0.334	-0.279	-0.516	-0.474	-0.427
		CL3	**0.917**	-0.436	-0.341	-0.594	-0.482	-0.481
Perceived Interaction	Conversation	CONV1	-0.351	**0.860**	0.433	0.556	0.526	0.484
		CONV2	-0.338	**0.894**	0.381	0.509	0.448	0.415
		CONV3	-0.445	**0.881**	0.501	0.675	0.405	0.532
		CONV4	-0.234	0.442	0.023	0.209	0.076	0.069
	No-Delay	DELAY1	-0.313	0.408	**0.915**	0.644	0.455	0.486
		DELAY2	-0.211	0.352	**0.798**	0.525	0.413	0.286
		DELAY3	-0.313	0.496	**0.929**	0.648	0.504	0.457
	Engaging	ENG1	-0.524	0.604	0.677	**0.814**	0.508	0.651
		ENG2	-0.481	0.605	0.471	**0.772**	0.524	0.695
		ENG3	-0.429	0.396	0.347	**0.701**	0.367	0.421
		ENG4	-0.337	0.321	0.451	**0.727**	0.321	0.413
		ENG5	-0.347	0.482	0.289	0.505	0.299	0.244
		ENG6	-0.354	0.509	0.699	**0.742**	0.507	0.590
		ENG7	-0.463	0.346	0.437	**0.744**	0.409	0.439
Self-Efficacy		SE1	-0.462	0.353	0.553	0.583	**0.872**	0.606
		SE2	-0.478	0.462	0.482	0.529	**0.902**	0.545

	SE3	-0.373	0.524	0.305	0.433	**0.856**	0.471
Satisfaction	SA1	-0.491	0.522	0.447	0.699	0.615	**0.981**
	SA2	-0.454	0.507	0.478	0.690	0.604	**0.981**

8.4 Structural Model

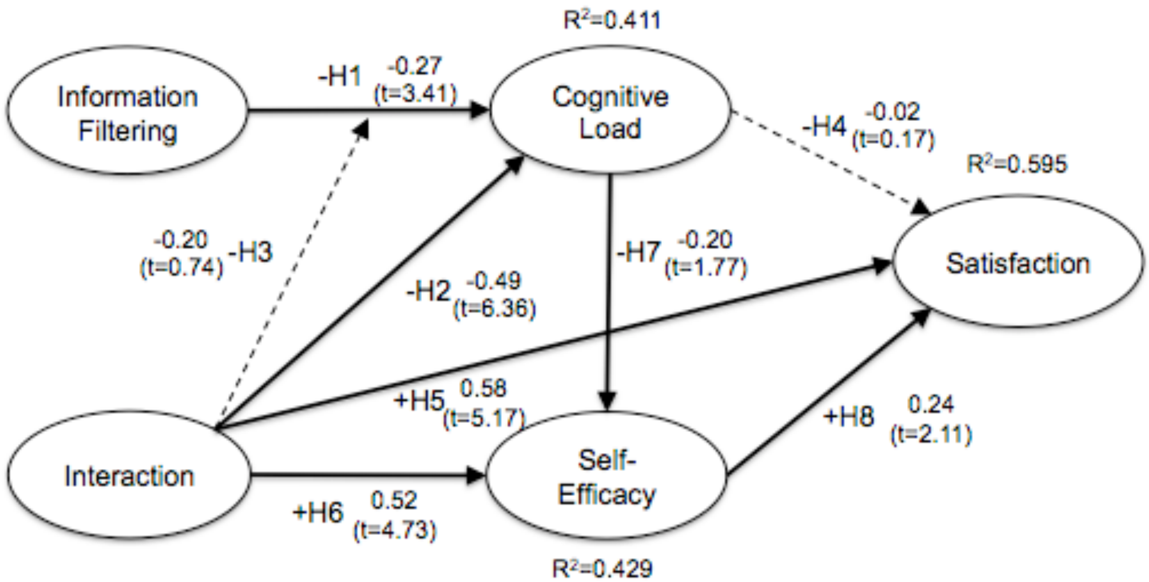


Figure 8: Structural Model

The structural model was assessed by examining path coefficients, their significance level, and the R^2 values. Perceived Interaction was treated as 1st order construct in the model[1]. Path coefficients indicate the strengths of the relationships between two constructs (Figure 8). R^2 values show the amount of variance explained by the independent constructs (Barclay et al. 1995). The final dependent construct, Satisfaction, has an R^2 value of 0.595, which indicates that the research model accounts for 59.5% of the variance in the dependent variable when Satisfaction was measured by accuracy. It is also instructive to examine the R^2 values for the intermediate variable in the structural model. The R^2 values for "Cognitive Load" and "Self-Efficacy" were 0.411

[1] We analyzed the model either Perceived Interaction as 1st order construct and 2nd order construct. The results of both analyses were similar.

and 0.429 respectively. R^2 of both Cognitve Load and Self-Efficacy were high enough to make the interpretation of the path coefficients meaningful.

Path coefficients in the structural model were computed with the entire sample. Bootstrapping with 500 resamples was computed to obtain the t-values corresponding to each path, as shown in Fig. 8. The acceptable t-values for one-tailed tests are 1.64 and 2.33 at the significance levels of 0.05 and 0.01. Information Filtering had a negative impact on Cognitive Load (β = -0.26, $p<0.01$), and therefore H1 was supported. Interaction had a negative impact on Cognitive Load (β = -0.49, $p<0.01$), supporting H2. Interaction effect of Information Filtering and Interaction on Cognitive Load had a negative association (β = -0.20, not significant), and H3 was not supported. Though, Cognitive Load had a negative impact on Satisfaction, it is not significant and therefore H4 was not supported (β = -0.02, not significant). Interaction had a positive impact on Satisfaction (β = 0.58, $p<0.01$), supporting H5. Interaction also had a positive impact on Self-Efficacy, (β = 0.52, $p<0.01$) supporting H6. Cognitive Load had a negative impact on Self-Efficacy (β = -0.20, $p<0.05$), and H7 was supported. Finally Self-efficacy had a positive impact on Satisfaction (β = 0.24, $p<0.05$) and therefore, H8 was supported.

8.5 Tests for Common Method Bias

We conducted two types of statistical analyses to assess the threat of common methods bias: Harman's one factor test and latent variable test (Podsakoff et al. 2003). First, in Harman's one factor test, the emergence of a single factor that accounts for a large proportion of the variance in factor analyses suggests a common methods bias. However, no such factor emerged. We loaded all items used to measure both independent and dependent variables into a single exploratory factor analysis. The analysis produced six factors with eigenvalues higher than 1. Taken together, these factors explained 80.7% of the variance in the data, with the first extracted factor accounting for 47.1% of the variance. Given that more than one factor was extracted from the analysis and the first factor accounted for less than 50% of the variance, common method bias is unlikely to be a significant issue.

Second, in a latent variable approach, we added a first-order factor with all of the measures in the theoretical model as indicators (Podsakoff et al. 2003). A common method factor was therefore added in the research model (Liang et al. 2007). The results presented in Appendix C demonstrate that the average substantively explained variance of the indicators is 0.713, whereas the average method-based variance is 0.054. The ratio of substantive variance to method variance is 13.2:1. Given the small magnitude and insignificance of method variance, common method bias is unlikely to be a serious concern in this study.

8.6 Limitations

Laboratory experimentation provides a highly controlled environment for hypothesis testing while it has a few methodological limitations. First, users had a limited time to use the prototype. The prototypes with interactive and information filtering features provided features which might be unfamiliar to some users. Users may have needed more time to get familiar with interactive and information filtering capabilities that were used in the study. Second, the experiment of this study is limited to the retrieval of use cases and does not examine its use on other contexts (Happel and Seedorf 2006). Finally, the use of student subjects may limit the generalizability of the results. However many prior studies on software engineering demonstrate that student subjects provided valid results (Runeson 2003; Singer and Vinson 2002).

9. DISCUSSION AND IMPLICATIONS

This study empirically confirms that information filtering and interaction have significant impact on cognitive load and self-efficacy. Interaction and self-efficacy are found to have significant impact on satisfaction. This study confirms the relationship between information filtering and cognitive load that was established in a study on ontology pruning (presented in Chapter 3). Six of the eight hypotheses were supported.

The interaction effect between information filtering and perceived interaction on cognitive load was not significant although it had a negative association with cognitive load. We posit that the interactive use of a well-organized ontology can reduce cognitive load thereby increasing user's satisfaction (i.e. hypotheses 3 and 4). However these hypotheses were not supported. Although the negative directions among constructs (i.e. interaction effect, cognitive load, and satisfaction) were found, they were not statistically significant. The statistical power of both hypotheses 3 and 4 was less than 0.3, which was weak to capture the hypothesized relationships. A stronger treatment or increased sample size may help address this issue.

Both perceived interaction and self-efficacy had significant impact on satisfaction. Perceived interaction had a positive impact on satisfaction. Perceived interaction also has an indirect impact on satisfaction via self-efficacy. Thus, Self-efficacy partially mediated the relationship between perceived interaction and satisfaction. The main effect between perceived interaction and satisfaction was statistically significant. The calculated effect size ((R^2 with mediator – R^2 with mediator)/(1 - R^2 with mediator) = (0.595-0.569)/(1-

0.595)) was weak (0.064) (Cohen 1988). However the mediation effect of self-efficacy was significant according to Sobel test (see Appendix D). This result suggests that an interactive system can directly increase user's satisfaction with the system. Indirect increase of user's satisfaction via self-efficacy is not strong. However we should interpret this weak intervening effect of self-efficacy with caution. Even though the intervening effect is weak, it is statistically significant. Therefore, system developers should develop interactive information retrieval systems that increase users' self-efficacy in order to increase user satisfaction. For example, providing positive feedback in a prompt and engaging way can increase users' belief about their capabilities to produce results and enhance user satisfaction.

Cognitive load has a negative impact on satisfaction. When users perceive high-levels of cognitive load, their satisfaction decreases. However, this relationship was not statistically significant. Prior research notes that users stop perusing a long list of retrieved items. Instead, 1) simply discard a large number of results, 2) restart a query in order to reduce the size of the retrieved result, and 3) examine only the top five or six results and select among them. Satisfaction can be maximized when cognitive load is minimized. When users receive a short list of items which are relevant, they will be satisfied with the system. Prior research on e-commerce systems suggests that customers abandon a site when presented with a lengthy list of items for perusal (Nielsen 2004). Therefore, the design of information systems which reduce cognitive load will lead to high user satisfaction. In our study, if the results provided by the system had been more compact and rank-ordered based on their relevance, it might have reduced the cognitive load and thereby increased satisfaction significantly.

The mediation effect of self-efficacy between cognitive load and satisfaction was minimal and not significant. The path coefficient values of H7 (-0.2) and H8 (0.24) were significant although the path coefficient value of H4 (-0.02) was not significant. The main effect between cognitive load and satisfaction was not significant. The mediation effect of self-efficacy was also weak and not significant. The calculated effect size is weak (0.084) (Cohen 1988). The result of Sobel test shows that the mediation effect of self-efficacy was not significant (Appendix D).

This result suggests that a large variation of satisfaction can be explained by a direct/mediation effect of self-efficacy. Self-efficacy directly affects satisfaction and works as a mediation variable between perceived interaction and satisfaction. Therefore, the developers of information retrieval systems should pay particular attention to factors such as interaction and cognitive load that affect perceived self-efficacy.

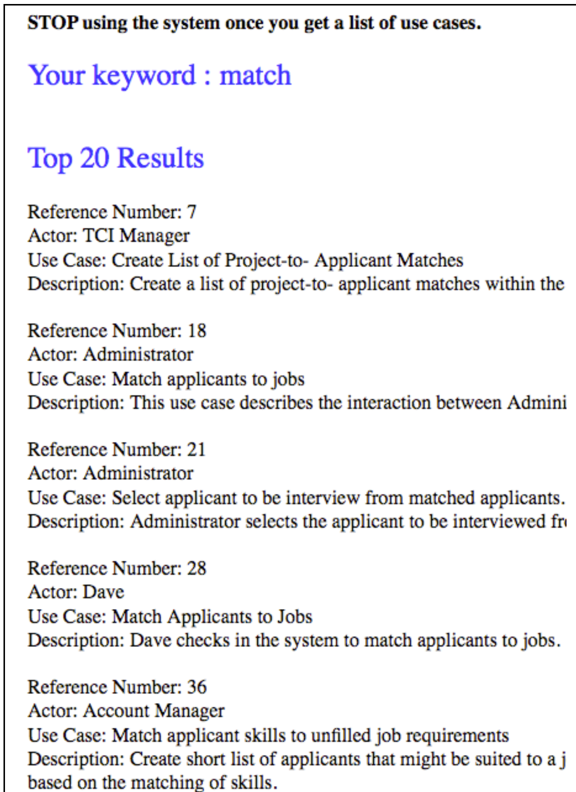
10. CONCLUSIONS AND FUTURE WORK

This research contributes to the literature on systems development in several ways. To facilitate the retrieval of use cases from a library and assess how user satisfaction is affected, this study develops a methodology and prototypes and empirically evaluates them with a theoretically grounded model. Our study suggests that a system which interacts with a user intelligently reduces cognitive load and increases self-efficacy and satisfaction. The model draws on interaction theory, which provides explanation on how interaction between the user and the system with ontological knowledge can increase self-efficacy and satisfaction.

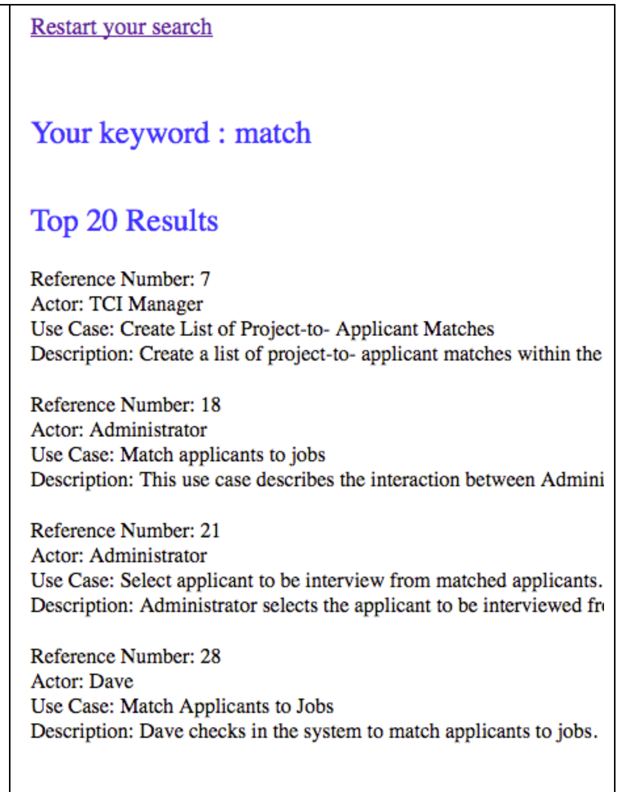
The proposed approach is implemented in a prototype. The interactive query system allows users to retrieve relevant use cases accurately, thereby enhancing the reuse of use cases in large and complex system development projects. An existing requirement management tools like RequisitePro (Rational 2005) which provides only a keyword based use case retrieval feature can benefit from incorporating this approach.

Topics for future research include the extension of our approach by using semantic information that will help infer the relationships among use cases. Also, our interactive approach using ontology can be extended to other critical tasks such as system maintenance and testing (Happel and Seedorf 2006). Further validation of the prototype is needed to assess its impact on task performance (say, the accuracy of results).

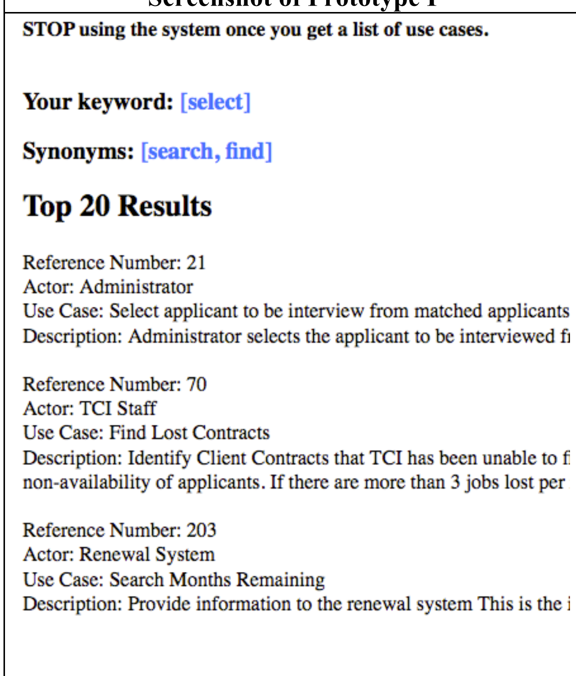
Appendix A: Screenshots of Four Prototypes



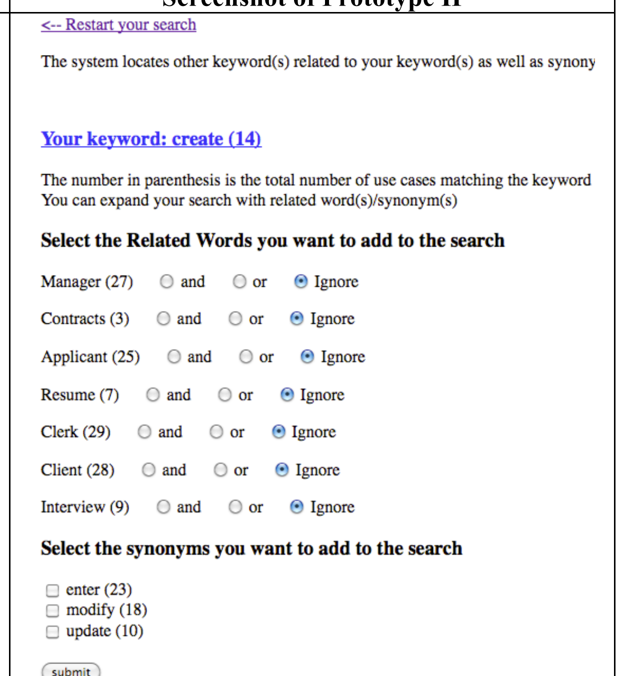
Screenshot of Prototype I



Screenshot of Prototype II



Screenshot of Prototype III



Screenshot of Prototype IV

Appendix B: Survey Questionnaire

Instruction

You are a systems analyst at TCI, a company that develops web based systems. You are working on a project for the development of a web based system that helps users find suitable dating partners. Your assignment is to develop use cases for the critical functionalities that should be supported by the system. You recognize that reviewing use cases from past projects that have included functionalities similar to your current project will be very helpful in your assignment. Your manager has provided you access to an online library of use cases from past projects completed at TCI. You can use a web based retrieval system that allows you to retrieve use cases from this library.

Your task in this experiment is to use this online retrieval system to find two of the most relevant use cases from the library. Each use case in the library is identified by a reference number. At the conclusion of your search, you need to report the reference numbers of the two most relevant use cases.

Specifically, your task involves the following steps:

1) Access the Use Cases Retrieval System (by clicking on the link at the end of these instructions).
2) Enter keyword(s) and click the Submit button to find use cases that match the keyword(s). - Do NOT use the system more than two times.
3) Identify the two of the most relevant use cases from the retrieved results.
4) Report the reference numbers of the two most relevant use cases in the text boxes provided below.

Online Use Cases Retrieval System (Depending on the treatment, one of the four prototypes is provided)

1. Report the reference numbers of the two most relevant Use Cases in the text boxes provided below.

______

______

2. Did you use the use cases retrieval system?

Yes No

3. Is the retrieval system interactive?

Yes No

4. Please take the survey below.

Cognitive load (Three items, Seven likert scale (1 = very little; 7 = very much))

- How much mental effort was required to perform the entire task (identifying keywords, using the system and selecting two use cases)?
- How difficult was it for you to perform the entire task (identifying keywords, using the system and selecting two use cases)?
- How burdensome was the task of identifying keywords, using the system and selecting two use cases?

Perceived Interaction (Fourteen items, Seven likert scale (1 = not at all descriptive; 7 = very descriptive)

Conversation (4 items)

- The retrieval system helps me INCREMENTALLY refine my search by adding more keywords.
- The retrieval system provides the ability to add ADDITIONAL keywords after displaying the results.
- The retrieval system is interactive.
- The retrieval system DOES NOT provide the ability to refine my search by adding more keywords.

No-Delay (3 items)

- The retrieval system provides fast response.
- The retrieval system responds slowly.
- The retrieval system operates at high speed.

Engaging (7 items)

- The retrieval system keeps my attention focused on the task.
- It is easy to select relevant use cases from the results provided by the system.
- The interaction with the retrieval system is unmanageable.
- The retrieval system DOES NOT allow me to keep my focus on the task.
- The retrieval system interacts with me passively rather than actively guiding me in my task.
- The retrieval system provides immediate answers to my search request.
- The retrieval system DOES NOT provide relevant use cases.

Self-efficacy (Three items, Eleven likert scale (0 = completely disagree; 10 = completely agree))

- I believe I have the ability to retrieve the most relevant use cases from the system.

- I believe I have the ability to INTERACTIVELY use the system by refining my search to find the most relevant use cases.
- I believe I have the ability to locate the most relevant use cases with ADDITIONAL relevant keywords provided by the system.

Satisfaction (Two items, Seven likert scale (1 = strongly disagree; 7 = strongly agree))

- All things considered, I am very satisfied with the retrieval system.
- Overall, my interaction with the retrieval system is very satisfying.

Please provide the following background information.

Age
Gender (Male/Female)
Past educational experience with systems analysis (in month)
Past professional experience with systems analysis (in month)
Past system development experience (in month)

My level of proficiency in using information retrieval system (such as google) is (seven likert scale, 1 = very low; 7 = very high)

Appendix C: Common Method Bias Analysis

Construct	Indicator	Substantive Factor Loading (R1)	$R1^2$	Method Factor Loading (R2)	$R2^2$
Ontology	Ontology	1	1.000	0	0.000
Perceived Interaction	CONV1	0.443	0.196	0.255	0.065
	CONV2	0.587*	0.345	0.065	0.004
	CONV3	0.849**	0.721	-0.079	0.006
	DELAY1	1.109**	1.230	-0.355*	0.126
	DELAY2	1.065**	1.134	-0.468*	0.219
	DELAY3	1.156**	1.336	-0.378*	0.143
	ENG1	0.603**	0.364	0.25	0.063
	ENG2	0.225	0.051	0.557*	0.310
	ENG3	0.392	0.154	0.203	0.041
	ENG4	0.858**	0.736	-0.25	0.063
	ENG6	0.819**	0.671	-0.043	0.002
	ENG7	0.454	0.206	0.198	0.039
Cognitive Load	CL1	0.901**	0.812	0.011	0.000
	CL2	1.007**	1.014	0.079	0.006
	CL3	0.852**	0.726	-0.09	0.008
Self-Efficacy	SE1	0.754**	0.569	0.136	0.018
	SE2	0.888**	0.789	0.019	0.000
	SE3	0.995**	0.990	-0.152	0.023
Satisfaction	SA1	0.969**	0.939	0.015	0.000
	SA2	0.993**	0.986	-0.015	0.000
Average		**0.806**	**0.713**	**-0.002**	**0.054**

$*p < .05$, $**p < .01$

Appendix D: Test of Mediation Effect of Self-Efficacy

Perceived Interaction -> **Self-Efficacy** -> Satisfaction

	Test Stat	Std. Error	p-value
Sobel	2.051	0.061	0.040
Aroian	2.016	0.062	0.044
Goodman	2.088	0.060	0.037

Cognitive Load -> **Self-Efficacy** -> Satisfaction

	Test Stat	Std. Error	p-value
Sobel	-1.251	0.033	0.147
Aroian	-1.374	0.035	0.169
Goodman	-1.542	0.031	0.123

REFERENCES

Agarwal, R., Sambamurthy, V., and Stair, R. "Research Report: The Evolving Relationship Between General and Specific Computer Self-Efficacy--An Empirical Assessment," *Information Systems Research* (11:4) 2000, p 418.

Anderson, E., and Sullivan, M. "The antecedents and consequences of customer satisfaction for firms," *Marketing science* (12:2) 1993, pp 125-143.

Back, J., and Oppenheim, C. "A model of cognitive load for IR: implications for user relevance feedback interaction," *Information Research* (6:2) 2001, pp 6-2.

Balzer, W.K., Doherty, M.E., and O'Connor, R.J.R. "Effects of cognitive feedback on performance," *Psychological bulletin* (106:3) 1989, pp 410-433.

Bandura, A. "Self-efficacy mechanism in human agency," *American Psychologist* (37:2) 1982, pp 122-147.

Bandura, A., and Adams, N. "Analysis of self-efficacy theory of behavioral change," *Cognitive Therapy and Research* (1:4) 1977, pp 287-310.

Barclay, D., Higgins, C., and Thompson, R. "The partial least square(PLS) approach to casual modeling: Personal computer adoption and use as an illustration," *Technology* (2:2) 1995, pp 285-309.

Bearden, W., Netemeyer, R., and Mobley, M. *Handbook of marketing scales* Sage Publ., 1993.

Berners-Lee, T., Hendler, J., and Lassila, O. "The Semantic Web," *Scientific American* (5) 2001, pp 1-19.

Billinghurst, M., and Starner, T. "Wearable devices: new ways to manage information," *Computer* (32:1) 1999, pp 57-64.

Branting, L. "Acquiring customer preferences from return-set selections," *Case-Based Reasoning Research and Development*) 2001, pp 59-73.

Chin, W.W. "The partial least squares approach to structural equation modeling," in: *Modern Methods for Business Research,* G.A.Marcoulides (ed.), Lawrence Erlbaum Associates, Mahwah, 1998, pp. 295-336.

Conesa, J., and Olivé, A. "A Method for Pruning Ontologies in the Development of Conceptual Schemas of Information Systems," *Journal on Data Semantics* (5) 2006, pp 64-90.

Conesa, J., Storey, V.C., and Sugumaran, V. "Using Semantic Knowledge to Improve Web Query Processing," in: *Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems*, Springer, Klagenfurt, Austria, 2006, pp. 106-117.

Cronin, J.J.J., and Taylor, S.A. "Measuring Service Quality: A Reexamination and Extension," *Journal of Marketing* (56) 1994, pp 55-68.

Cyc "Cyc Ontology," (http://www.cyc.com).

Delone, W.H., and McLean, E.R. "Information Systems Success: The Quest for the Dependent Variable," *Information Systems Research* (3:1, March) 1992, pp 60-95.

Driver, M. "Exploring student perceptions of group interaction and class satisfaction in the web-enhanced classroom," *The Internet and Higher Education* (5:1) 2002, pp 35-45.

Drolet, A., and Frances Luce, M. "The rationalizing effects of cognitive load on emotion-based trade-off avoidance," *Journal of Consumer Research* (31:1) 2004, pp 63-77.

Fantechi, A., Gnesi, S., Lami, G., and In, A.M. "Application of Linguistic Techniques for Use Case Analysis," IEEE Joint International Conference on Requirements Engineering, 2002, pp. 157-164.

Fantechi, A., Gnesi, S., Lami, G., and Maccari, A. "Applications of linguistic techniques for use case analysis," *Requirements Engineering* (8:3), Aug 2003, pp 161-170.

Fornell, C., and Larcker, D. "Evaluating structural equation models with unobservable variables and measurement error," *Journal of marketing research*) 1981, pp 39-50.

Gopher, D., and Braune, R. "On the psychophysics of workload- Why bother with subjective measures?," *Human Factors* (26) 1984, pp 519-532.

Gruber, T.R. "Toward principles for the design of ontologies used for knowledge sharing," in: *Formal Ontology in Conceptual Analysis and Knowledge Representation,* N. Guarino and R. Poli (eds.), Kluwer Academic Publishers, Deventer, The Netherlands, 1993.

Guarino, N. "Formal Ontology and Information Systems," in: *Proceedings of the FOIS'98*, IOS Press, 1998, pp. 3-15.

Happel, H., and Seedorf, S. "Applications of ontologies in software engineering," Citeseer, 2006, pp. 5-9.

Henry, J., and Stone, R. "A structural equation model of end-user satisfaction with a computer-based medical information system," *Information Resources Management Journal* (7) 1994, pp 21-21.

Hogarth, R.M. *Judgment and Choice* John Wiley & Sons, 1996.

Johnson, R., and Marakas, G. "Research report: the role of behavioral modeling in computer skills acquisition: toward refinement of the model," *Information Systems Research* (11:4) 2000, pp 402-417.

Kaiya, H., and Saeki, M. "Ontology Based Requirements Analysis: Lightweight Semantic Processing Approach," Fifth International Conference on Quality Software (QSIC'05), 2005, pp. 223-230.

Keller, J., and Suzuki, K. "Learner motivation and e-learning design: A multinationally validated process," *Learning, Media and Technology* (29:3) 2004, pp 229-239.

Kettinger, W.J., and Lee, C.C. "Perceived service quality and user satisfaction with the information services function," *Decision Sciences* (25:5,6) 1994, pp 737-766.

Klischewski, R. "Ontologies for e-document management in public administration," *Business Process Management Journal* (12:1) 2006, pp 34-47.

Lamport, M.A. "Student-Faculty Informal Interaction and the Effect on College Student Outcomes: A Review of the Literature," *Adolescence* (28:112) 1993.

Lassila, O., and McGuinness, D. "The Role of Frame-Based Representation on the Semantic Web," KSL-01-02, Stanford University, Stanford, California, p. 9.

Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A., and Cohen, M.D. "Intelligent information-sharing systems," *Communications of the ACM* (30:5) 1987, pp 390-402.

Marcia, B., and Pulman, S. "A method for controlling the production of specifications in natural language," *The Computer Journal* (38) 1995, pp 310-318.

Mason, O. "QTag - a Portable POS Tagger," Accessed on Jan 7, 2005 (http://www.english.bham.ac.uk/staff/omason/software/qtag.html).

Matuszek, C., Cabral, J., Witbrock, M., and DeOliveira, J. "An Introduction to the Syntax and Content of Cyc," 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering 2006.
McMillan, S., and Hwang, J. "Measures of perceived interactivity: An exploration of the role of direction of communication, user control, and time in shaping perceptions of interactivity," *Journal of Advertising* (31:3) 2002, pp 29-42.
McSherry, D. "Balancing user satisfaction and cognitive load in coverage-optimised retrieval," *Knowledge-Based Systems* (17:2-4) 2004, pp 113-119.
Mitchell, T., Hopper, H., Daniels, D., George-Falvy, J., and James, L. "Predicting self-efficacy and performance during skill acquisition," *Journal of Applied Psychology* (79:4) 1994, pp 506-517.
NE, F., and U, S. "Reasoning in Attempto Controlled English," Proceedings of the International Workshop on Principles and Practice of Semantic Web Reasoning, 2003, pp. 174-188.
Nielsen, J. *Designing web usability* Pearson Education, 2004.
Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., and Musen, M.A. "Creating Semantic Web contents with Protege-2000," *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]* (16:2) 2001, pp 60-71.
Oliver, R. "Varieties of value in the consumption satisfaction response," *Advances in Consumer Research* (23) 1996, pp 143-147.
Paas, F. "Training strategies for attaining transfer of problem-solving skill in statistics: a cognitive-load approach," *Journal of educational psychology* (84:4) 1992, pp 429-434.
Paas, F., Tuovinen, J., Tabbers, H., and Van Gerven, P. "Cognitive load measurement as a means to advance cognitive load theory," *Educational Psychologist* (38:1) 2003, pp 63-71.
Parasuraman, A., Zeithaml, V., and Berry, L.L. "Reassessment of Expectations as a Comparison Standard in Measuring Service Quality: Implications for Further Research," *Journal of Marketing* (58:January) 1994, pp 111-124.
Park, S., Kim, H., Ko, Y., and Seo, J. "Implementation of an efficient requirments-analysis supporting system using similarity measure techniques," *Information & Software Technology* (42:6) 2000, pp 429-438.
Pintrich, P., and Schrauben, B. "Studentsí motivational beliefs and their cognitive engagement in classroom academic tasks," *Student perceptions in the classroom*) 1992, pp 149-183.
Podsakoff, P., MacKenzie, S., Lee, J., and Podsakoff, N. "Common method biases in behavioral research: A critical review of the literature and recommended remedies," *Journal of Applied Psychology* (88:5) 2003, pp 879-903.
Poli, R. "Descriptive, formal and formalized ontologies," *International Journal of Human-Computer Studies* (56:6), June 2002 2002, pp 639-664.
Rational, I.B.M. "Rational RequisitePro," 2005 (http://www3.software.ibm.com/ibmdl/pub/software/rational/web/datasheets/version6/reqpro.pdf).
Runeson, P. "Using students as experiment subjectsñan analysis on graduate and freshmen student data," Citeseer, 2003.

Saeki, M. "Ontology-Based Software Development Techniques," in: *ECRIM News*, 2004, pp. 14-15
Sengupta, K., and Te'eni, D. "Cognitive Feedback in GDSS: Improving Control and Convergence," *MIS Quarterly* (17:1) 1993, pp 87-113.
Singer, J., and Vinson, N. "Ethical issues in empirical studies of software engineering," *IEEE Transactions on Software Engineering*) 2002, pp 1171-1180.
Storey, V.C., Burton-Jones, A., Sugumaran, V., and Purao, S. "CONQUER: A Methodology for Context-Aware Query Processing on the World Wide Web," *Information Systems Research* (19:1) 2008, pp 3-25.
Sutton, D.C. "Linguistic Problems with Requirements and Knowledge Elicitation " *Requirements Engineering* (5:2) 2000, pp 114-124.
Sweller, J. "Cognitive load during problem solving: Effects on learning," *Cognitive Science* (12) 1988, pp 257-285.
Sweller, J., and Chandler, P. "Why Some Material Is Difficult to Learn," *Cognition and Instruction* (12:3) 1994, pp 185-233.
Venkatesh, V. "Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model," *Information Systems Research* (11:4) 2000, pp 342-365.
Venkatesh, V., and Davis, F. "A Model of the Antecedents of Perceived Ease of Use: Development and Test*," *Decision Sciences* (27:3) 1996, pp 451-481.
W3C "OWL Web Ontology Language," 2004 (http://www.w3.org/TR/owl-features/).
Wells, J.D., Fuerst, W.L., and Choobineh, J. "Managing information technology (IT) for one-to-one customer interaction," *Information & Management* (35:1) 1999, pp 53-62.
Wixom, B., and Todd, P. "A theoretical integration of user satisfaction and technology acceptance," *Information Systems Research* (16:1) 2005, pp 85-102.
Wood, R., Atkins, P., and Tabernero, C. "Self-efficacy and strategy on complex tasks," *Applied Psychology* (49:3) 2000, pp 430-446.
Wood, R., and Bandura, A. "Impact of conceptions of ability on self-regulatory mechanisms and complex decision making," *Journal of personality and social psychology* (56:3) 1989, pp 407-415.
Yi, M., and Davis, F. "Developing and validating an observational learning model of computer software training and skill acquisition," *Information Systems Research* (14:2) 2003, pp 146-169.
Yi, M., and Hwang, Y. "Predicting the use of web-based information systems: self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model," *International Journal of Human-Computer Studies* (59:4) 2003, pp 431-449.