Cognitive support for semi-automatic ontology mapping

by

Sean M. Falconer
Bachelor of Computer Science, University of New Brunswick, 2003
Master of Computer Science, University of New Brunswick, 2005

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

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

Cognitive support for semi-automatic ontology mapping

by

Sean M. Falconer
Bachelor of Computer Science, University of New Brunswick, 2003
Master of Computer Science, University of New Brunswick, 2005

Supervisory Committee

Dr. Margaret-Anne Storey, (Department of Computer Science)
Supervisor

Dr. Hausi A. Müller, (Department of Computer Science)
Departmental Member

Dr. Jens H. Weber-Jahnke, (Department of Computer Science)
Departmental Member

Dr. Francis Lau, (Department of Health and Information Science)
Outside Member

Supervisory Committee

Dr. Margaret-Anne Storey, (Department of Computer Science)
Supervisor

Dr. Hausi A. Müller, (Department of Computer Science)
Departmental Member

Dr. Jens H. Weber-Jahnke, (Department of Computer Science)
Departmental Member

Dr. Francis Lau, (Department of Health and Information Science)
Outside Member

ABSTRACT

Structured vocabularies are often used to annotate and classify data. These vocabularies represent a shared understanding about the terms used within a specific domain. People often rely on overlapping, but independently developed terminologies. This representational divergence becomes problematic when researchers wish to share, find, and compare their data with others. One approach to resolving this is to create a mapping across the vocabularies. Generating these mappings is a difficult, semi-automatic process, requiring human intervention. There has been little research investigating how to aid users with performing this task, despite the important role the user typically plays. Much of the research focus has been to explore techniques to automatically determine correspondences between terms.

In this thesis, we explore the user-side of mapping, specifically investigating how to support the user's decision making process and exploration of mappings. We combine data gathered from theories of human inference and decision making, an observational case

study, online survey, and interview study to propose a cognitive support framework for ontology mapping. The framework describes the user information needs and the process users follow during mapping. We also propose a number of design principles, which help guide the development of an ontology mapping tool called COGZ. We evaluate the tool and thus implicitly the framework through a case study and controlled user study.

The work presented in this thesis also helps to draw attention to the importance of the user role during the mapping process. We must incorporate a "human in the loop", where the human is essential to the process of developing a mapping. Helping to establish and harness this symbiotic relationship between human processes and the tool's automated process will allow people to work more efficiently and effectively, and afford them the time to concentrate on difficult tasks that are not easily automated.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

NCBO:	National Center of Biomedical Ontology
KIF:	Knowledge Interchange Format
OBO:	Open Biomedical Ontologies
OIL:	Ontology Inference Layer
(DAML)+OIL:	DARPA Agent markup language
OKBC:	Open Knowledge Base Connectivity
OWL:	Web Ontology Language
RDFS:	Resource Description Framework Schema
XOL:	XML-Based Ontology Exchange Languagee
EMF:	Eclipse Modeling Framework
MDE:	Model Driven Engineering
PIM:	Platform Independent Models
PSM:	Platform Specific Models
UML:	Unified Modeling Language
ATL:	Atlas Transformation Language
QVT:	Query View Transformation
EODM:	Eclipse Ontology Definition Metamodel

Acknowledgement

First, I would like to thank my supervisor Dr. Margaret-Anne (Peggy) Storey. Coming into my PhD, my background was largely theoretical, but through you, I was gradually introduced to the challenge of understanding users and designing tools that help support their processes. This is an exceptionally exciting and ever changing field that I am extremely grateful to now be a part of. Your enthusiasm and imagination were constant inspiration throughout my degree and to the entire CHISEL research group. I am forever indebted to you for your guidance.

I would also like to thank all the members of the CHISEL research group. Working in the lab was the greatest work experience of my professional and academic career. The support, ideas, and imagination of every group member helped contribute to my work and made everyday in the office an enjoyable and unforgettable experience.

I am also very grateful to my committee members: Dr. Hausi A. Müller, Dr. Jens H. Weber-Jahnke, and Dr. Francis Lau. Your suggestions both during my proposal and thesis presentation helped guide the construction of this work. In addition, Dr. Philip A. Bernstein, thank you for agreeing to be my external examiner. Your advice and knowledge of the subject area was invaluable.

I would like to acknowledge Oleg Golubitsky and Dmitri Maslov. Both of you helped introduce a shy computer science undergraduate to the world of computer science research and I am forever grateful to you. Your intelligence and desire to solve challenging problems has always inspired me and I am proud to now be your colleague and friend.

Mom, Dad, Jessica, and Sarah, thank you for your support. All of you have always been patient with listening to my academic ramblings and in exchange, I have always been your computer technical support. Each of you can always make me laugh and I love you all.

Finally, none of this would be possible without my wife and best friend Theresa. You have always been there for me and I cannot imagine this degree or this life without you.

Dedication

to my parents

Part I

The problem

Chapter 1

Introduction

Biomedical researchers and scientists use structured terminologies such as classification systems and ontologies to annotate and enrich the semantics of their data. The data may be descriptions of clinical trials, genes, experiments, or research papers. However, these scientists often work independently from each other and rely on different domain-specific terminologies. Comparing, sharing, and finding these different "pockets" of related research is very challenging. Relationships between similar terms in these heterogeneous terminologies have to be specified in order to facilitate data integration and sharing. Mapping, the process of relating these terms based on a shared meaning, is a very difficult task, one that relies on a combination of tool and algorithm development, along with human intervention.

The mapping process is difficult because terminologies are developed by humans and as a result they often encode our biases, cultural differences, and subjective world views. For example, we have witnessed heated debates among biologists over what a phenotype really means even though every first year biology text appears to clearly define this term. There is a great deal of complexity in determining conceptual matches. Humans struggle with categorizing and classifying certain types of data [Mur02], e.g., is a tomato a fruit or a vegetable?

Categorizing and relating data requires a human to be in the loop, whereby the human can use their real world knowledge and domain expertise to make these important decisions. However, much of the research on mapping or relating data has been focused on the precision and recall of automated procedures for discovering correspondences. Despite years of research on this topic, coping with data heterogeneity is still one of the most time-consuming data management problems. According to Bernstein *et al.* [BM07], every database research self-assessment has listed interoperability of heterogeneous data as one of the main research problems.

1.1 Motivation

Ontologies are one approach to representing structured terminologies or "knowledge". They provide a shared and common understanding about a specific domain [DF02]. They represent the concepts and the various relationships within a domain. Ontologies are richer in structure than a taxonomy, as relationships between concepts are not restricted to containment or subclass relationships. They can also include part of, has a, and other domain-specific relationships.

Mapping ontologies is key to data and information integration [NGM08]. A mapping represents a relationship between instances of two data representations [Mel04]. For ontologies, this generally consists of matching synonymous terms/concepts between two ontologies. Mappings can be used to help search applications via query expansion, where a search query can be expanded using synonymous terms based on recorded mappings. Mappings can also be used to relate data, where a researcher may annotate their data with concepts from one ontology, but be able to relate their data to previous research annotated with concepts from a different ontology.

Since mapping ontologies is so vital to resolving data heterogeneity problems, it has received an increasing amount of attention in recent years. Mapping contests exist to compare the quality of ontology matchers [OAE06], a mapping API that specifies a format for expressing alignments has been proposed [Euz06], and workshops have been organized to discuss this problem [OM206]. However, the research emphasis has primarily been on the automation of this process, even though most ontology mappings involve the user at some

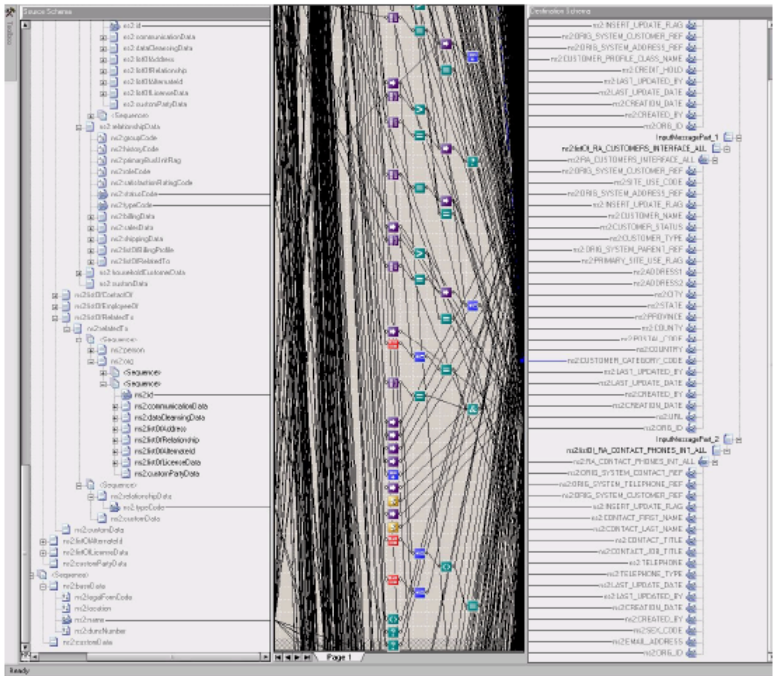

Figure 1.1. *Example of a mapping where the user interface does not scale [RCC05a].*

stage of the process.

Research in this area has largely ignored the issue of user intervention (with a few exceptions [RCC05a, MHH00]). Research has instead focused on designing tools and algorithms to compute candidate correspondences. Many of these tools provide only text file dumps of potential correspondences (e.g., FOAM [ES05]) or interfaces that quickly become unmanageable (see Figure 1.1). The responsibility of verifying and working through the mass of data computed by these algorithms is left to the user. This can be extremely difficult, requiring tremendous patience and an expert understanding of the ontology domain, terminology, and semantics.

Contrary to this existing research trend, we feel that since the human is critical to the success of the mapping procedure it follows that as researchers interested in addressing the problem of mapping, we must address and emphasize the human needs. We believe that this begins with understanding existing mapping processes, difficulties with using existing tools, and the user decision making process. Through this understanding, better tools can be developed that help rather than hinder users. *Cognitive support* can be introduced to the tools to reduce the *cognitive load* experienced by users. Cognitive support refers to the introduction of external aids to support cognitive processes [Wal02a], while cognitive load refers to the load on working memory during problem solving [PRS$^+$94, p. 710]. In agreement with Bernstein *et al.* [BM07], we believe that cognitive support, and hence better user interfaces, is critical to the biggest productivity gains in mapping tasks, not the improvement of precision and recall in matching algorithms.

1.2 Problem statement and research objectives

In this thesis, we focus on understanding what cognitive support means in the context of ontology mapping. Specifically, we address the problem: *How can users be supported during semi-automatic ontology mapping such that the accuracy and efficiency for creating mappings is improved?* Based on this problem, we address several specific research objectives:

O1: Determine implications for tool design based on biases and limitations of human inference.

O2: Determine which parts of the mapping task are difficult and which are simple.

O3: Determine which tools are being used and how they meet or do not meet user requirements.

O4: Discover the process users follow for constructing mappings.

O5: Discover opportunities for cognitive support in ontology mapping systems.

O6: Use the opportunities to create design elements that are necessary for supporting users during the mapping process.

O7: Create a tool that is based on the cognitive support design elements.

O8: Evaluate the tool and thus its design.

1.3 Approach and methodology

We approach the problem of improving user support for semi-automatic ontology mapping through four primary stages. First, we examine, through a series of user studies and background literature, which factors are important for ontology mapping, which problems users are experiencing, which process is currently being followed, and which tools they are using (Chapters 3, 4, 5, 6, and 7). Second, we combine results from these experiments and existing literature to propose a cognitive support framework for ontology mapping (Chapter 8). The framework consists of a number of user information needs as well as describes an ontology mapping process model. The framework identifies the various opportunities for cognitive support within mapping systems. We use these opportunities to formalize a set of mapping tool design requirements. Third, we use these requirements to develop an interactive semi-automated ontology mapping tool (Chapter 9). The final stage is the evaluation of the tool. We demonstrate that the approach is scalable to large biomedical ontologies, that it improves the accuracy and efficiency of mapping, and that it has been adopted by other researchers (Chapter 10). These stages form an iterative cycle, that is, the results from tool development and evaluation helps to inform our study phase and framework throughout the research.

We follow primarily a qualitative research methodology for the development of the framework. This is due to the exploratory nature of research objectives **O1** through **O5**. We also base the framework in part on literature from cognitive psychology and specifically on three behavioural experiments. For the initial evaluation of the framework, we follow qualitative evaluation procedures as outlined by Creswell [Cre03] and we evaluate the tool using a mixed methods approach [Cre03, p. 18]. These are discussed in more detail below in Section 1.5.

1.4 Scope

In this thesis, we limit our data integration scope to ontologies, however we believe that many of the problems inherent in this domain are consistent across other similar domains. Also, we focus primarily on specifying mappings between ontology concepts for the purpose of determining semantic equivalence. This is the type of mapping primarily supported in the biomedical community [BP, FBS, NGM08, UML]. Some applications of mappings, such as query translation and structured data integration, need more specific transformation rules in order to be carried out. We do not focus on this type of data integration, however, we do discuss how we have been able to adapt our technique to support this process (see Chapter 11).

1.5 Evaluation

We evaluate our framework following qualitative evaluation procedures. Specifically, we use triangulation [Cre03] to validate emergent themes. This involves verifying that the themes are present in multiple experiments and data sources. This provides justification or evidence that the theme is a consistent usage pattern across a population of users. The framework is also validated by expert reviews by publishing papers on the framework as well as using input from our colleagues at the National Center for Biomedical Ontology (NCBO) project [NCB].

The tool is evaluated following a mixed methods approach. We first demonstrate through a case study that our approach is scalable and feasible for large biomedical ontologies. We then evaluate, through a controlled lab study, that the tool makes significant improvements to the accuracy and efficiency of a user's evaluation process when constructing mappings. Finally, we discuss adoption from researchers and industry.

1.6 Contributions

This thesis makes several contributions to the ontology research community. The studies discussed in this thesis are the first studies specifically investigating human inference for mapping, how users interact with mapping tools, which processes they follow and how they interact in teams. The results of these experiments are combined to form a cognitive support framework that describes the information needs of mapping users, the process they follow and a set of design principles for developing mapping tools. This framework provides requirements for any researcher interested in developing mapping tools.

The requirements were used to develop our own mapping tool called COGZ, which combines visualization and filtering techniques to help support the user's decision making process. The evaluation experiment we introduce is the first study specifically investigating the cognitive support provided by a mapping tool. The findings from the study help contribute to a theory of required tool support. All of these results have helped to draw attention to the important role the user fulfills during the mapping process. We have helped to emphasize that improvements to mapping quality and adoption will arise when users are more effectively supported with the constructing mappings.

1.7 Organization of the thesis

This thesis is organized into three parts: The problem (Chapters 1 and 2), theory building (Chapters 3 through 8), and applying and evaluating the framework (Chapters 9 through 12). See Figure 1.2 for an overview of the outline.

In Chapter 2, we introduce relevant background material related to ontologies, ontology mapping, and the current state of the art in this field. Following this, in Chapter 3, we discuss related work from cognitive psychology on human inference and decision making. We use this to suggest several implications for ontology mapping, which later helps guide the development of our cognitive support framework. In Chapter 4, we discuss related

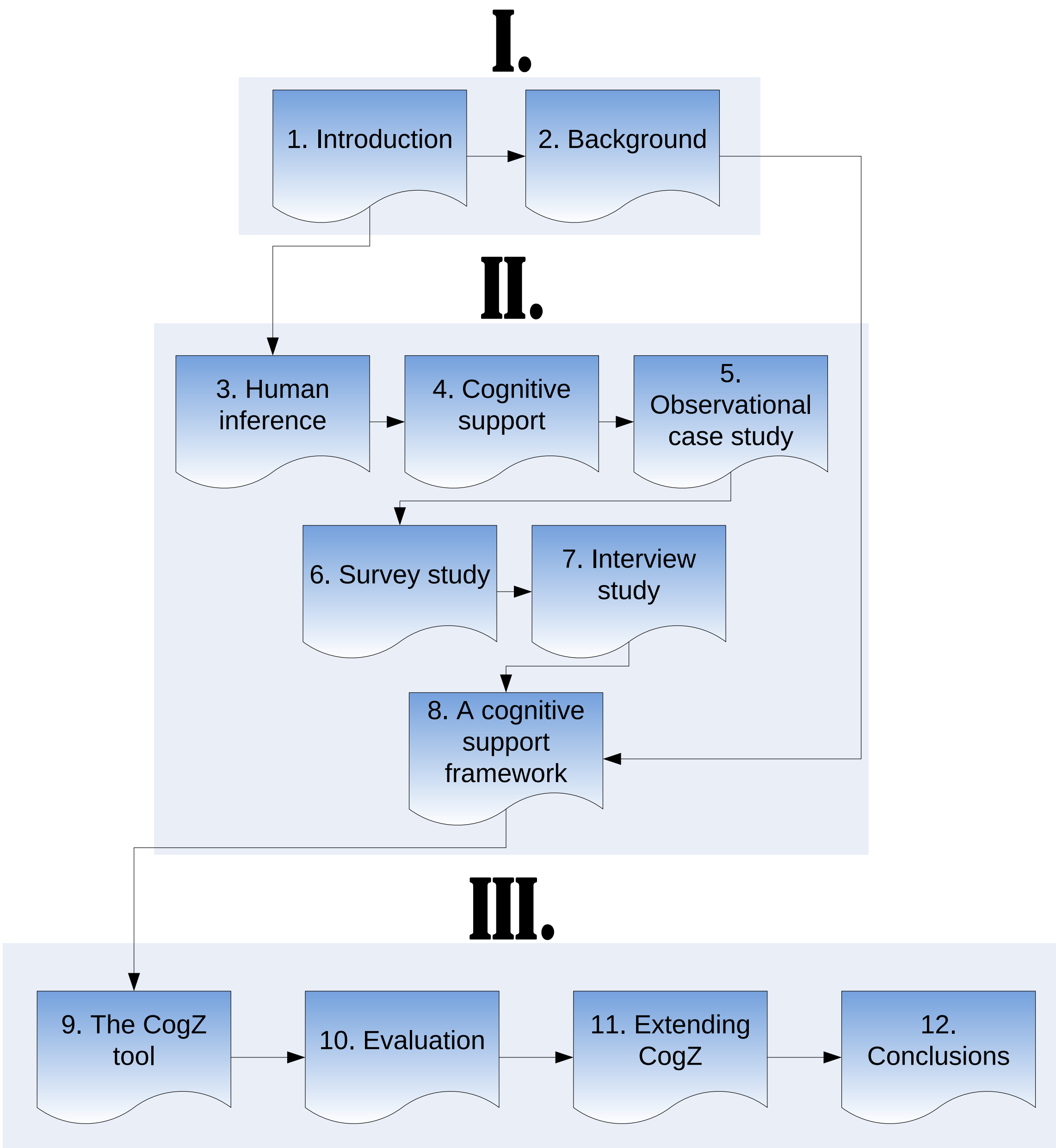

Figure 1.2. *Thesis outline*

work on *cognitive support*, specifically work by Walenstein. His work fits well with the implications derived in Chapter 3. Next, in Chapters 5, 6, and 7, we present three different exploratory studies investigating the user-side of mapping. In Chapter 8, we combine the results from these studies along with the relevant work on human inference to introduce a cognitive support framework for ontology mapping. We use this framework to help guide the development of a tool (Chapter 9), which we evaluate in Chapter 10. We discuss some of the extensions made to COGZ in Chapter 11 and finally discuss our future work and contributions in Chapter 12.

Chapter 2

Ontologies and the mapping problem

This chapter presents a brief history of the term "ontology" and how it has been adopted by computer science. We also introduce the mapping problem in the context of ontologies and a motivating example. A brief survey of existing tools for mapping ontologies is presented along with a description of standard approaches for automatically computing mappings between ontologies.

2.1 What is an ontology?

The word *ontology* is generally thought to have originated in early Greece from Plato and Aristotle [Gru09]. The earliest known record of the word is from 1606 in the Latin form *ontologia* [Lor06, ØSU05], while the earliest English occurrence of ontology appeared in Bailey's dictionary in 1721 [Bai21, Cor08]. In philosophy it is *the study of being or existence* [Gru09].

In Computer Science, ontology was first adopted by researchers in Artificial Intelligence (AI) in the early 1980s [McC80]. The AI community primarily used the term to refer to a theory of a modeled world or part of a knowledge system. Later, in the early 1990s, in an effort to create interoperability standards, an "ontology layer" was introduced as a standard component for a knowledge system technology stack [NFF+91]. Shortly afterwards, an ontology was famously defined by Tom Gruber as an "explicit specification of a conceptualization" [Gru93]. The introduction of this definition, although quite controversial, is credited with ontology becoming a technical term within Computer Science [Gru09].

Ontology in Computer Science shares commonalities with the philosophical origins. In both cases, an ontology is the representation of objects, concepts, and other entities, along with the properties and relations that hold between them [Gru93]. However, the focus of the two areas is different. Philosophers are concerned with debating how to construct an ontology and the entities of reality, while the focus in Computer Science is on developing controlled vocabularies and the practical uses of an ontology [ØAS05]. In Computer Science, ontologies are primarily developed for the purpose of knowledge sharing and reuse [GPP+93].

While only small ontologies have been developed in philosophy, a large number of ontologies have been developed by computer scientists and in the physical sciences. For example, BioPortal [BP], an online application for sharing and navigating ontologies, contains over 100 ontologies specifically related to the biomedical field. Approximately 100 ontologies from a variety of domains are listed in the Protégé Ontology Library [PT2] and Swoogle [SG2], the ontology search engine, contains over 10,000 ontologies in its index.

2.2 Components of an ontology

Ontologies consist of a number of different components that are used to help define and model a domain. In this section, we provide a brief overview of the languages that have been developed to create ontologies. We focus primarily on the Web Ontology Language, and present a short description of its primary components.

There are many different languages available for developing ontologies; some of these include: CycL [LG89, Len95], Protégé frames, Knowledge Interchange Format (KIF) [GF92], Open Biomedical Ontologies (OBO) [OBO], Ontology Inference Layer (OIL) [Hor00], DARPA Agent markup language (DAML)+OIL [Hv01], Open Knowledge Base Connectivity (OKBC) [CFF+98], Web Ontology Language (OWL) [OWLb], Resource Description Framework Schema (RDFS) [BG00], and XML-Based Ontology Exchange Language

(XOL) [KCT]. The purpose of an ontology is consistent across all of these languages: it helps to define the concepts, relationships, and other distinctions relevant for modeling a domain [Gru09]. The languages usually have different degrees of formality and granularity. Many of the languages evolved from earlier languages.

Ontologies are often classified as *lightweight* or *heavyweight* [Tun07]. A lightweight ontology consists of concepts, relationships between the concepts and properties that describe the concepts. This view of ontology is similar to software and database schema modeling. A heavyweight ontology contains more explicit constraints and axioms to help define the intended meaning of a concept.

Recently, the adoption of ontologies has increased in both research and industry, especially as interest and development in the *Semantic Web* has continued. The Semantic Web vision is that the Internet can be a globally linked database, one that supports data interoperability and machine readable semantics [Pal01]. It is primarily about two important things: using common formats for data integration and a language for specifying how data relates to real world objects [SW]. Ontologies are a large component of the Semantic Web. They can be used to specify a common language and multiple applications can use concepts from the same ontology. This ensures that each application is "talking" about the same thing, potentially making data integration easier. Ontologies are part of the technology stack for the W3C Semantic Web standard [BLHL01]. The W3C also recommends OWL [OWLb] as a standard for developing ontologies for the Semantic Web[1].

The OWL standard has three different "flavours": *OWL Lite*, *OWL DL*, and *OWL Full*. OWL Lite is a subset of OWL DL and is intended for users primarily needing a classification hierarchy and simple constraints [OWLa]. OWL DL is a subset of OWL Full and is intended for users that want maximum expressiveness while still having guaranteed decidability. In OWL DL, all OWL language constructs are available, but they can only be used under certain restrictions. Also, OWL DL supports Description Logic [BCM+03]. Finally, OWL Full gives users maximum expressiveness and freedom with defining their ontology,

[1] In this thesis, most ontologies discussed were developed using OWL.

but reasoning with the ontology is not guaranteed to be tractable. OWL ontologies are specified using XML.

The three primary components of OWL are *classes*, *properties*, and *individuals*. Classes are the building blocks of OWL; they are the concepts or terms within the domain being modeled. Properties describe relationships between classes and individuals, where an individual is a member of a class. For example, we can define a class "Country", which may have properties like "name", "population", and "GDP", and a member or individual of this class could be the instance "Canada".

2.3 The mapping problem

A generic mapping problem occurs when there exists different representations of similar information. These representations can be physical, like text, pictures, or events that we experience. They can also be our own mental representations of these physical objects and events. A mapping must be constructed in order to transform one representation into another.

In computer science, a mapping problem is often described in terms of mapping two *schemas*. A schema is an expression that defines a set of possible instances [BM07], like an ontology or database schema. There are two main categories of mapping generation [BM07]. First, given a source and target schema, a user or tool defines mappings between the two data representations. This is the common category of mapping typically associated with applications of ontologies, XML and database schemas. Second, given only one schema, a second schema is derived (semi-)automatically according to some metamodel, along with the mapping. Database persistence tools such as Hibernate [HIB] use this approach to semi-automatically convert an object model into a relational model.

Ontology mapping is a solution to the semantic heterogeneity problem [SE08]. A mapping solution consists of a set of correspondences between semantically related entities of ontologies. Formally, a correspondence is defined as a 5-uple: $\langle id, e_1, e_2, n, r\rangle$, where id is

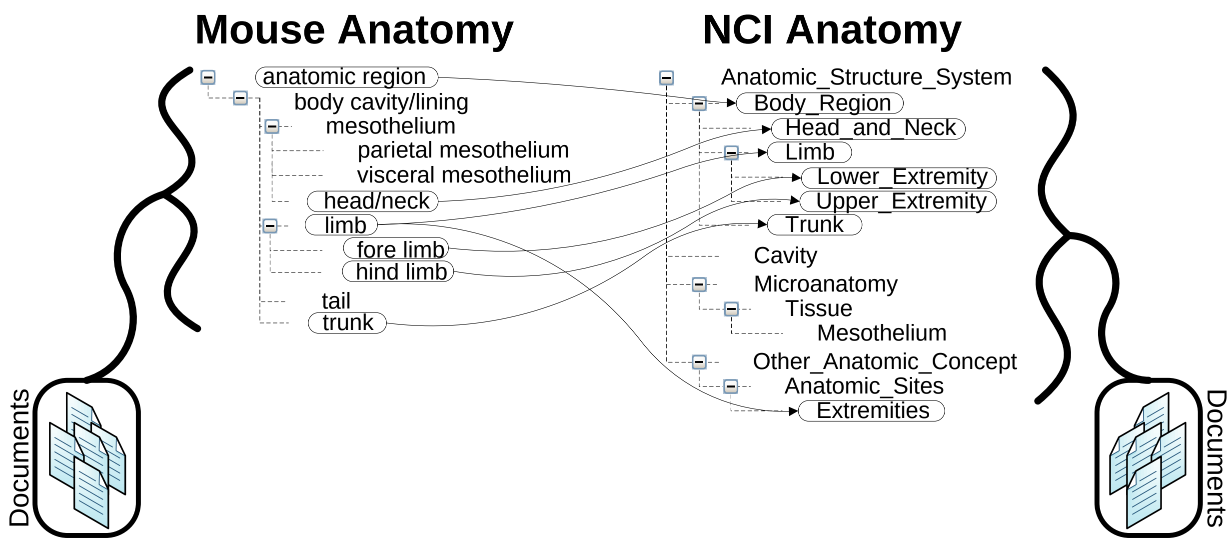

Figure 2.1. *Example mapping between the Mouse Adult Gross Anatomy ontology and NCI Anatomy. Terms from both the source and target ontology involved in the mapping are bounded by the rounded rectangles and mapping correspondences are represented by the solid curved arcs. Two separate document repositories have been annotated with terms from the ontologies.*

a unique identifier of the given correspondence, e_1 and e_2 are entities from the source and target ontologies respectively, n is the confidence measure that the correspondence holds for e_1 and e_2, and r is the relation [Euz06, SE05]. Relations typically include equivalence ($=$), more general ($\sqsupseteq$), disjointness ($\perp$), and overlapping ($\sqcap$) [SE08], although the exact relationships specified are often application dependant. Also, the confidence value may be omitted depending on the goals for producing the mapping.

2.3.1 Motivating example

Consider the two partial ontologies shown in Figure 2.1. On the left, referred to here as the *source*, a partial branch of the Mouse Adult Gross Anatomy[2] (MA) ontology is shown and on the right, referred to here as the *target*, a partial branch of the National Cancer Institute (NCI) Thesaurus[3] ontology is shown. In this scenario, both ontologies have been

[2] http://bioportal.bioontology.org/ontologies/38664
[3] http://bioportal.bioontology.org/ontologies/13578

used as a controlled vocabulary to annotate collections of scientific documents. For example, a biomedical curator may have associated terms from the source, like “mesothelium”, “limb”, and “trunk”, with text from research papers stored in a document repository. These annotations can be used to categorize, explore, and search the document collection.

The two ontologies contain many of the same concepts, but the concepts are sometimes represented differently (e.g., “fore limb” and “Lower_Extremity”). This heterogeneity poses a problem if scientists familiar with the MA ontology wish to search documents from the target (NCI Thesaurus) document repository. To resolve the potential terminological differences, a mapping can be constructed between the two ontologies. The mapping correspondences can then be used in a search or navigation application so that terms from the source ontology can find matching documents within the target repository.

In Figure 2.1, a partial mapping between the two branches are represented by the bounded terms with arrows mapping a source term to a target term. The mapping correspondences can potentially be used for other applications besides search. For example, the mappings are the first step towards merging the two ontologies into a single ontology or transforming data represented by one ontology into the other. However, constructing these mappings is a difficult process. In the next section, we expand on why this is such a difficult problem.

2.3.2 Why is mapping difficult?

The study of mapping problems is pervasive throughout computing. In theoretical computer science, the problem manifest itself in areas like graph matching [Kuh55], string matching [SM97, p. 49], and complexity analysis [GJ79, p. 13]. In the database community, this problem appears in the form of different database versions, similar databases developed independently, and the construction of mappings between object to relational models. As we introduce new technologies and seemingly new research areas, this problem manifests itself yet again. We see it in XML schema mapping [biz07], report generation [cry], and Extract-Transform-Load (ETL) tools [etl].

As pervasive as this problem is within Computer Science, it is even more pervasive in biological information processing. Humans deal with mapping problems everyday. Writing, reading, and interpreting our surroundings, are all forms of mapping. When we see the world with our eyes, we must transform this information into our own internal representation. This transformation process is quite natural for us, but still relies on mapping one representation to another. For example, both the construction and interpretation of a cave drawing is a mapping problem. To construct such a drawing, the artist first witnessed or experienced some event that he internalized in his head. This interpretation was then externalized in a pictorial form.

Conceptually, ontology mapping is closely related to these "real world" problems of mapping. The conceptualization that is specified in an ontology is an interpretation of real world entities that exist as abstract ideas or as mental symbols in our "heads". With ontologies, we attempt to encode and define these concepts. Ontologies are supposed to help alleviate some of the problems of heterogeneity because if a concept is formally defined then we know exactly what that concept means and mapping it to synonymous concepts should be easier. However, these definitions have limits that are intrinsic to the ontology representational formalism.

The formalism for defining terms in an ontology is based on the *classical view* of categories. This view proposes that definitions are the proper way to characterize meaning and category membership [Mur02, p. 11]. This view was first proposed by Aristotle and was later adopted in early psychological approaches to understanding concepts. The classical view makes three major claims [Mur02, p. 15]. The first is that concepts are mentally represented definitions and a definition provides the necessary and jointly sufficient conditions for membership in the category. The second is that every object is either in or not in a category. The third is that all category members are equally good, that is, a member of a category cannot be a more typical member than another member of that category.

Since Rosch's work in the 1970s [Ros78], this view has mostly disappeared in cognitive psychology. This is in part due to philosophical and empirical reasons. One of the main philosophical arguments against the definitional approach is that it is very difficult to define concepts through necessary and sufficient conditions. Wittgenstein used the example "dog" to make this argument [Mur02, p. 17]. For example, we can define a dog as a four legged animal, that barks, has fur, eats meat, etc. However, this is not a valid definition, there are dogs with less than four legs and there are also hairless dogs. Another problem is that the neatness of the classical view does not appear to match human concepts. People have difficulty assessing category membership and studies have shown that people are not able to segregate items into clear members and non-members [Ham79].

Despite these advances in cognitive psychology, ontologies are still based on this classical view of categories. OWL even retains the use of the terms necessary and sufficient conditions. This approach for ontology construction is attractive because definitions can be described using logical expressions, which are then machine processable. However, we cannot rely purely on the definitions to solve issues of heterogeneity. The definitions cannot encapsulate the real world knowledge that the domain experts possess. Thus, it is critical to understand the domain and the context in which a term is intended.

Obtaining this understanding is very difficult. Languages are known to be *locally ambiguous*, meaning that a sentence may contain an ambiguous portion that is no longer ambiguous once the whole sentence is considered [PPP]. Humans use detailed knowledge about the world to infer unspoken meaning [NLP]. As of yet, it is very difficult for machines to simulate this process.

The underlying data format used for specifying the ontology also introduces potential problems. The language used (e.g., OWL, RDF, XSD) constrains the expressiveness of the data representation. For example, many formats lack information relating to units of measure or intended usage [BM07]. Also, in ontologies, the concepts are largely characterized by a term or a small set of terms, which due to language may lack sufficient information to be properly interpreted.

Ontologies are also developed for different purposes and by users with potentially opposing world views. This may result in two ontologies describing concepts with different

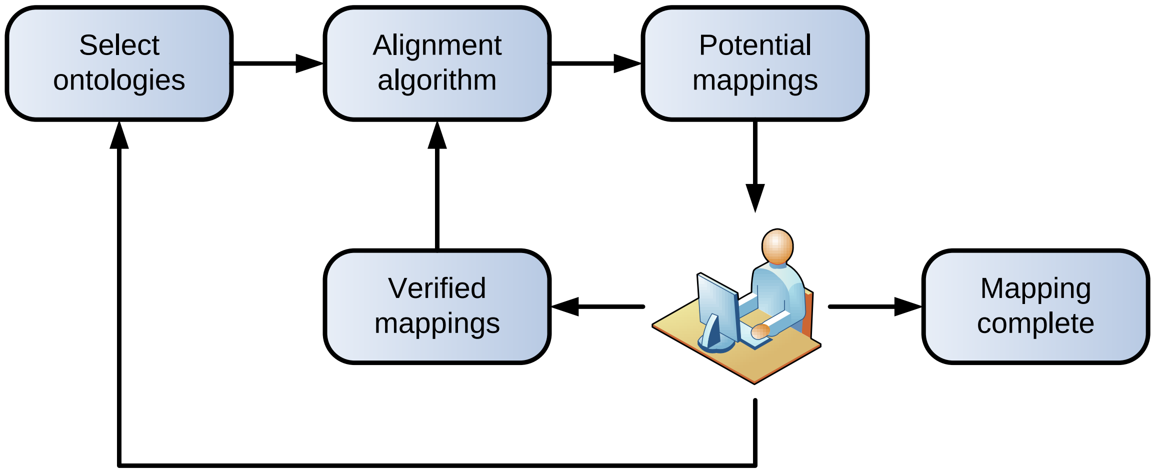

Figure 2.2. *Example of semi-automatic mapping process. A user is involved in iteration with the tool. As the user evaluates potential correspondences, their decisions are used by the tool to make other suggestions about mappings. This iteration continues until the user determines the mapping is complete.*

levels of granularity or the same concept with different intended application or meaning.

All of these issues make discovering and defining mappings a very challenging problem. In the next section, we discuss some of the various research tools and algorithm approaches that have been developed to help address this problem.

2.4 Ontology mapping tools

Ontology mapping is a prerequisite for many ontology-related applications. These include instance mediation across web sites, agent communication over the Internet, web service integration, and query and answer rewriting [dP04, Ee04]. The quality of these applications depends largely on the underlying mapping.

Most mappings are created semi-automatically, where a user works directly with a tool or inspects and manipulates output produced by a tool. Often, the user works in "iteration" with the tool; that is, as the user approves and rejects suggested correspondences, that information is used by the automated procedure to make further suggestions (see Figure 2.2).

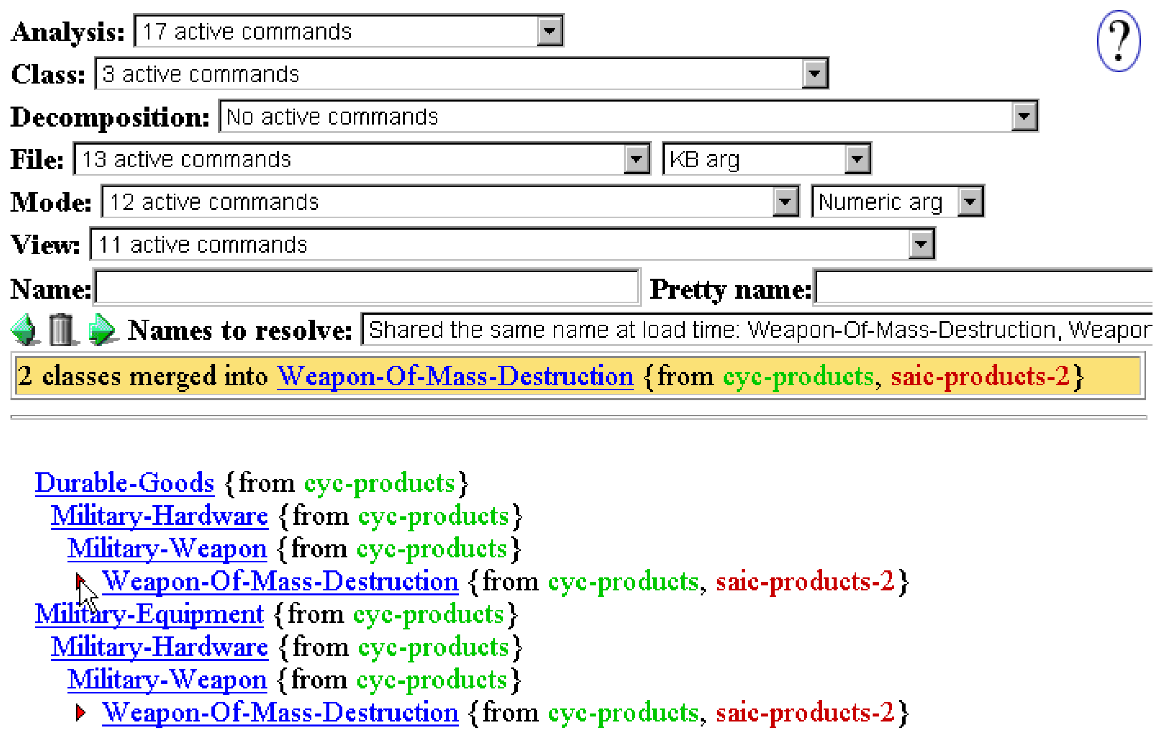

Figure 2.3. *Screenshot of Chimaera interface for merging two classes.*

A large variety of mapping tools exist to help compute correspondences. Most of their user-interfaces fall into one of three categories: console-based, web-based, and graphical user interfaces (some tools support more than one of these interfaces).

FOAM (Framework for Ontology Alignment and Mapping) [ES04b] is a tool for fully or semi-automatically aligning two or more OWL ontologies. The underlying alignment algorithm uses heuristics to compute similarity between ontological terms and the individual entities (concepts, relations, and instances). The authors of FOAM originally attempted to apply existing alignment algorithms, but found that the existing techniques, when applied to real-world datasets and use cases, did not meet their requirements [ES04b].

The software is available in two forms, as a downloadable Java application and also as a web service. The Java application only supports a console-based interface. The user supplies the application with a parameter file that specifies the location for the ontologies to align, an optional file of pre-known correspondences, and algorithm specifications.

The FOAM tool saves all the computed correspondences to a results file, in the form: "<uri1>;<uri2>;<confidence>", where the <confidence> is a number between zero and one representing how strong the matching is between <uri1> and <uri2>. In the semi-automatic approach, FOAM asks the user to verify certain correspondences and the user can specify in the parameter file the maximum number of questions that should be posed.

Chimaera [MFRW00] is a software system that supports ontology merging and diagnosis. The system has a web-based interface where the user interacts with web forms to upload ontologies, select algorithm parameters, and merge similar ontology entities (see Figure 2.3). The merge algorithm produces a candidate list of correspondences as matching terms, based on term name similarity, term definitions, possible acronyms and expanded forms, and suffix matching [Ee04]. Similar to FOAM, Chimaera supports OWL ontologies and produces mapping correspondence results in OWL descriptions.

Two other related tools are MoA Shell [Ins03] and the OWL Ontology Aligner [Zhd]. MoA was developed by the Electronics and Telecommunication Research Institute (ETRI) in South Korea and is an environment for merging ontologies [ES04b]. There is currently not a lot of detailed information about how MoA works, although it is known that its mapping algorithms are similarity based. MoA exposes a library of methods via a console-based interface. The environment only supports OWL files. Similarly, the OWL Ontology Aligner only supports OWL files, but uses a web-based interface. The user supplies the URIs to the two ontologies to map in a web form, and the system produces a list of possible mapping correspondences in HTML formatted table.

COMA++ [Do06], PROMPT [NM03], AlViz [LS06], OLA [ELTV04], and the NeOn toolkit all support graphical user interfaces. COMA++ automatically generates mappings between source and target schemas (XML or OWL), and draws lines between potentially matching terms (see Figure 2.4). Users can also define their own term matches by interacting with the schema trees. Hovering over a potential correspondence displays a confidence level about the match as a numerical value between zero and one.

PROMPT (see Figure 2.5) was developed by the Stanford Medical Informatics group

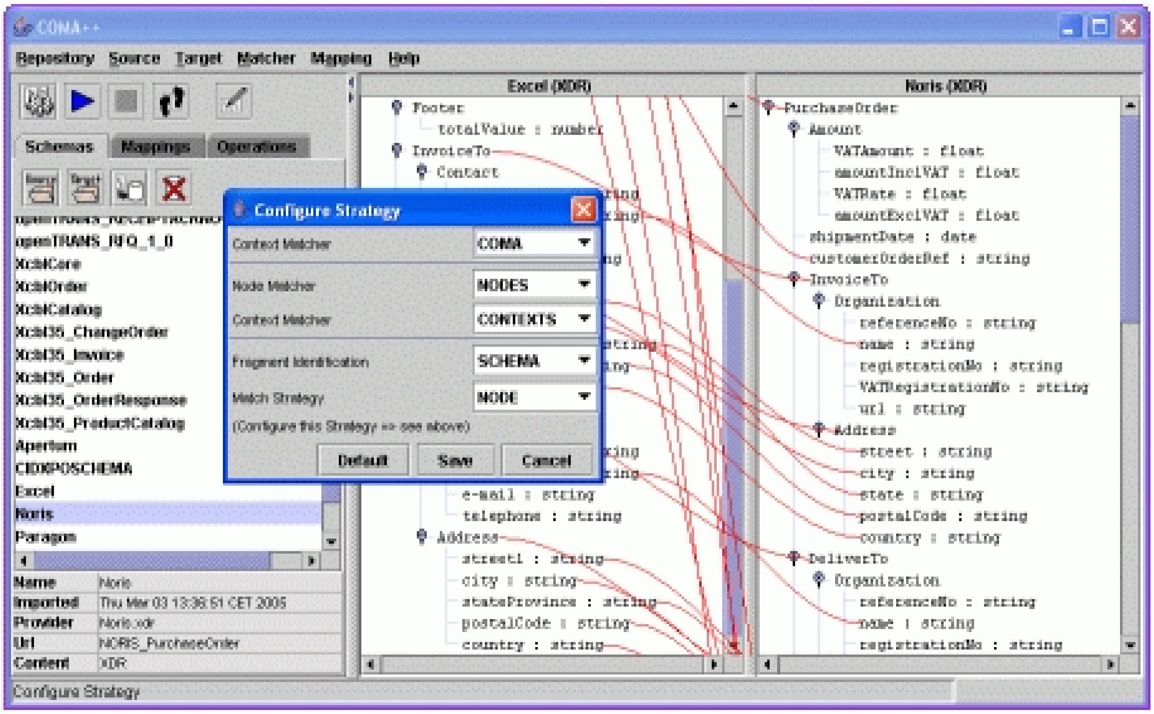

Figure 2.4. *Screenshot of COMA++ interface.*

and was built as a plugin for the popular ontology editor Protégé[4]. The plugin supports tasks for managing multiple ontologies including ontology differencing, extraction, merging, and mapping. PROMPT begins the mapping procedure by allowing the user to specify a source and target ontology. It then computes an initial set of candidate correspondences based largely on lexical similarity between the ontologies. The user then works with this list of correspondences to verify the recommendations or create custom correspondences that were missed by the algorithm. Once a user has verified a correspondence, PROMPT's algorithm uses this to perform structural analysis based on the graph structure of the ontologies. This analysis usually results in further correspondence suggestions. This process is repeated until the user determines that the mapping is complete. PROMPT saves verified correspondences as instances in a *mapping ontology* [CM03]. The mapping ontology provides a framework for expressing transformation rules for ontology mappings. It describes the source and target correspondence components and can also associate metadata with

[4] http://protege.stanford.edu

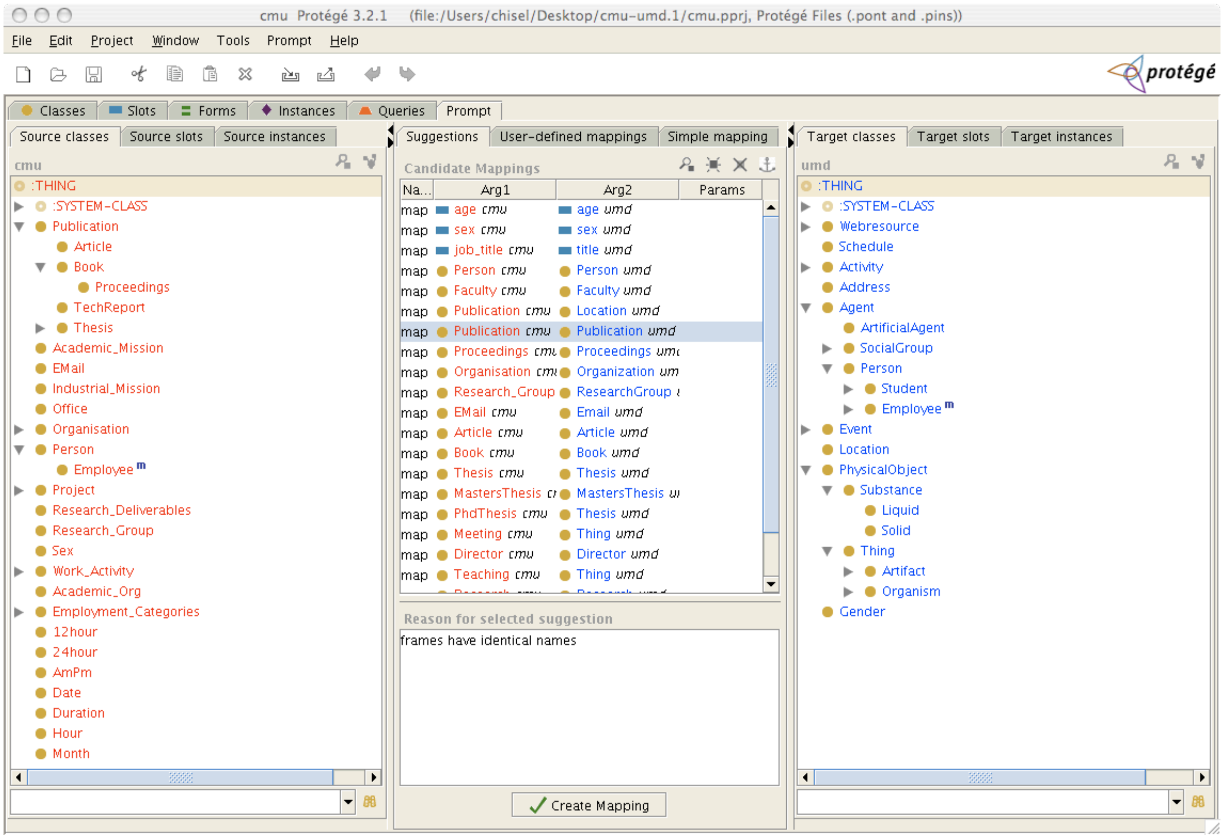

Figure 2.5. *Screenshot of* PROMPT *plugin while mapping two university ontologies.*

the correspondence, such as the date, who created the correspondence, and a user-defined comment.

Similar to PROMPT, AlViz is a plugin for Protégé, however the tool is primarily in an early research phase. AlViz was developed specifically for visualizing ontology alignments. It applies multiple-views via a cluster graph visualization along with synchronized navigation within standard tree controls (see Figure 2.6). The tool attempts to facilitate user understanding of the ontology alignment results [LS06] by providing an overview of the ontologies in the form of clusters. The clusters represent an abstraction of the original ontology graph; moreover, clusters are colored based on their potential concept similarity with the other ontology.

OLA (OWL Lite Alignment) is a tool for automated alignment construction as well

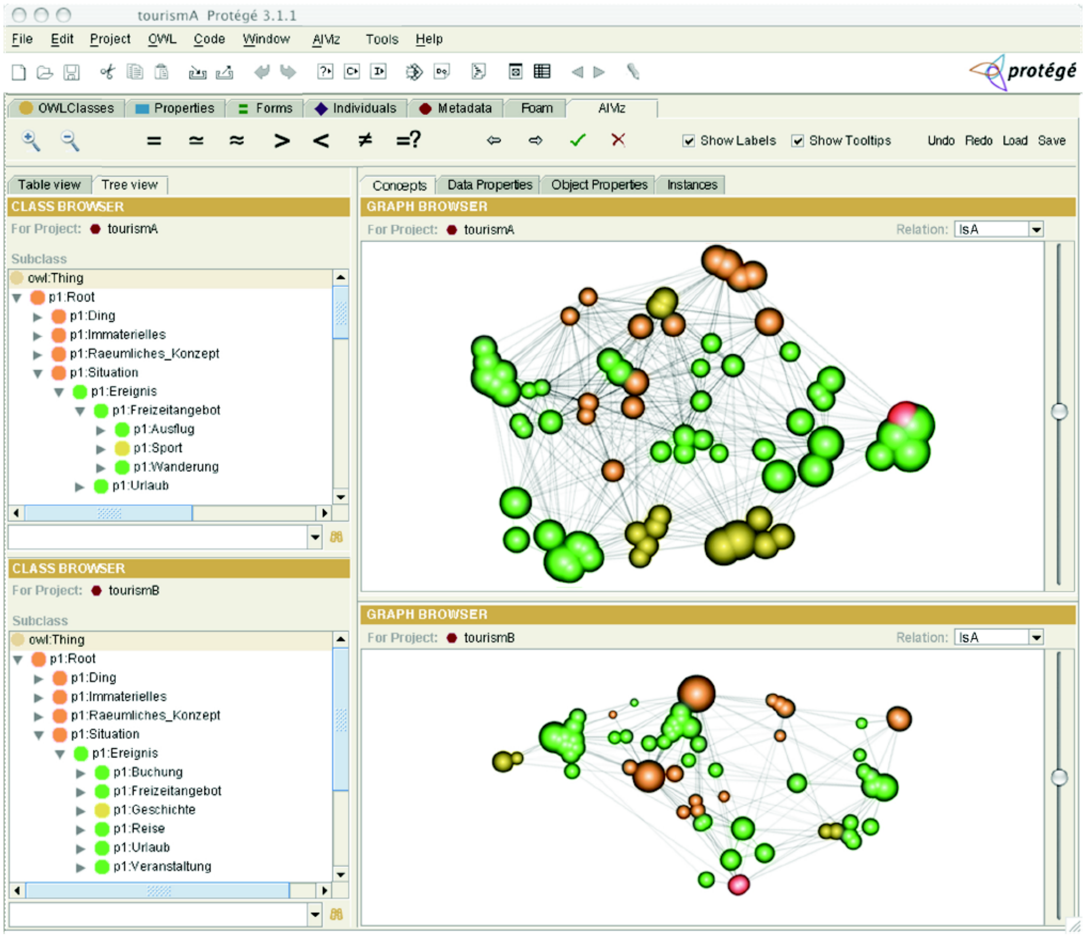

Figure 2.6. *Screenshot of AlViz plugin while mapping two tourism ontologies [LS06].*

as an environment for manipulating alignments [ELTV04]. The tool supports parsing and visualization of ontologies, automated computing of similarities between ontology entities, manual construction of alignments, visualization of alignments, and comparison of alignments (see Figure 2.7). OLA only supports OWL Lite ontologies and uses the Alignment API specified in [Euz06] to describe a mapping. The mapping algorithm finds correspondences by analyzing the structural similarity between the ontologies using graph-based similarity techniques. This information is combined with label similarity measures (e.g., Euclidean distance, Hamming distance, substring distance) to produce a list of mapping correspondences.

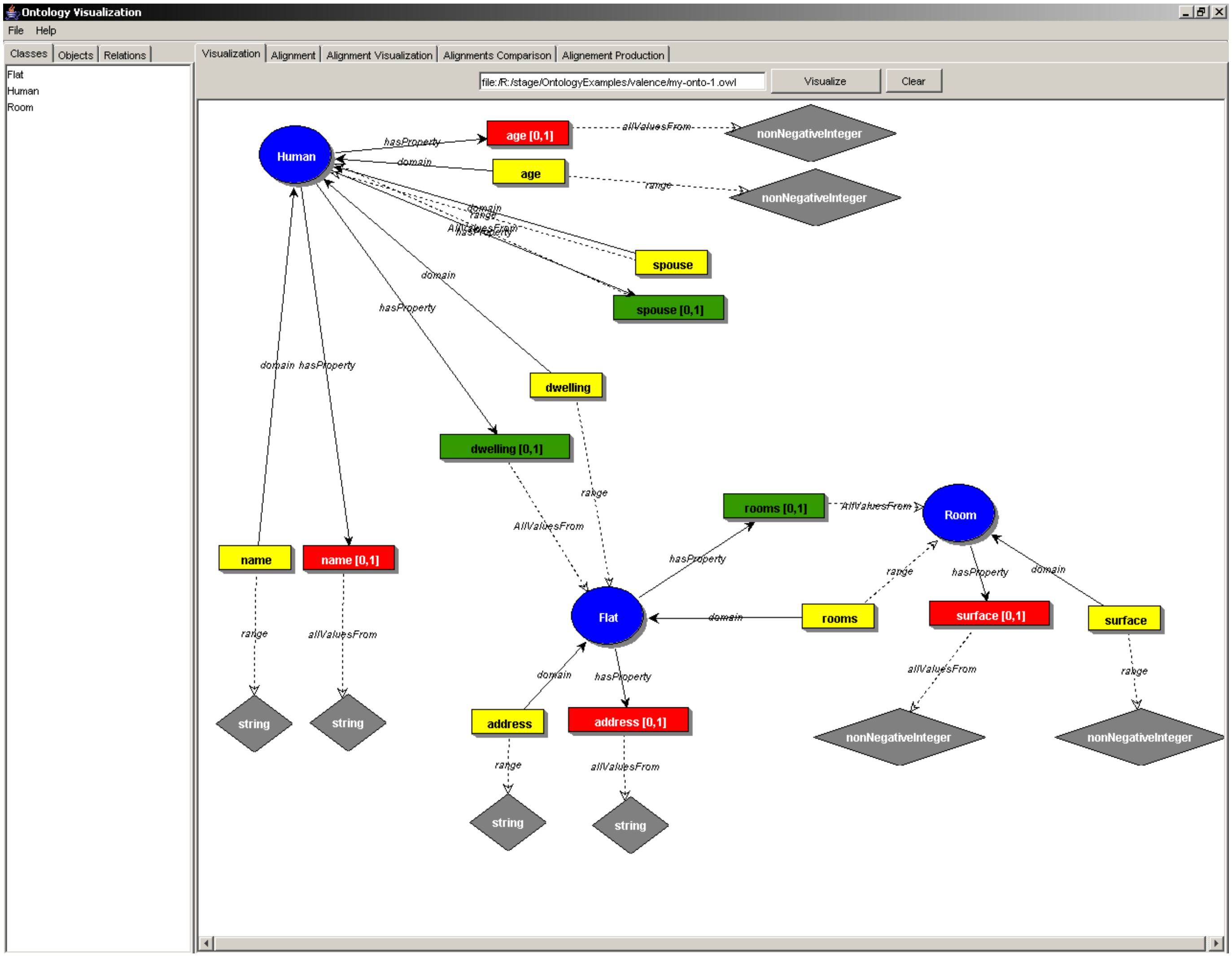

Figure 2.7. *Screenshot of OLA visualization of an OWL ontology.*

The NeOn toolkit [DdB+08], developed as an Eclipse plugin[5], is an environment for managing ontologies within the NeOn project[6]. NeOn supports run time and design time ontology mapping support and can be extended via plugins. The toolkit includes a mapping editor called OntoMap, which allows a user to create and edit alignments (see Figure 2.8). Similar to the previously mentioned tools, NeOn supports OWL ontologies, however it also supports RDF and F-Logic. The toolkit can also convert a variety of sources (e.g.,

[5] http://www.eclipse.org
[6] http://www.neon-project.org

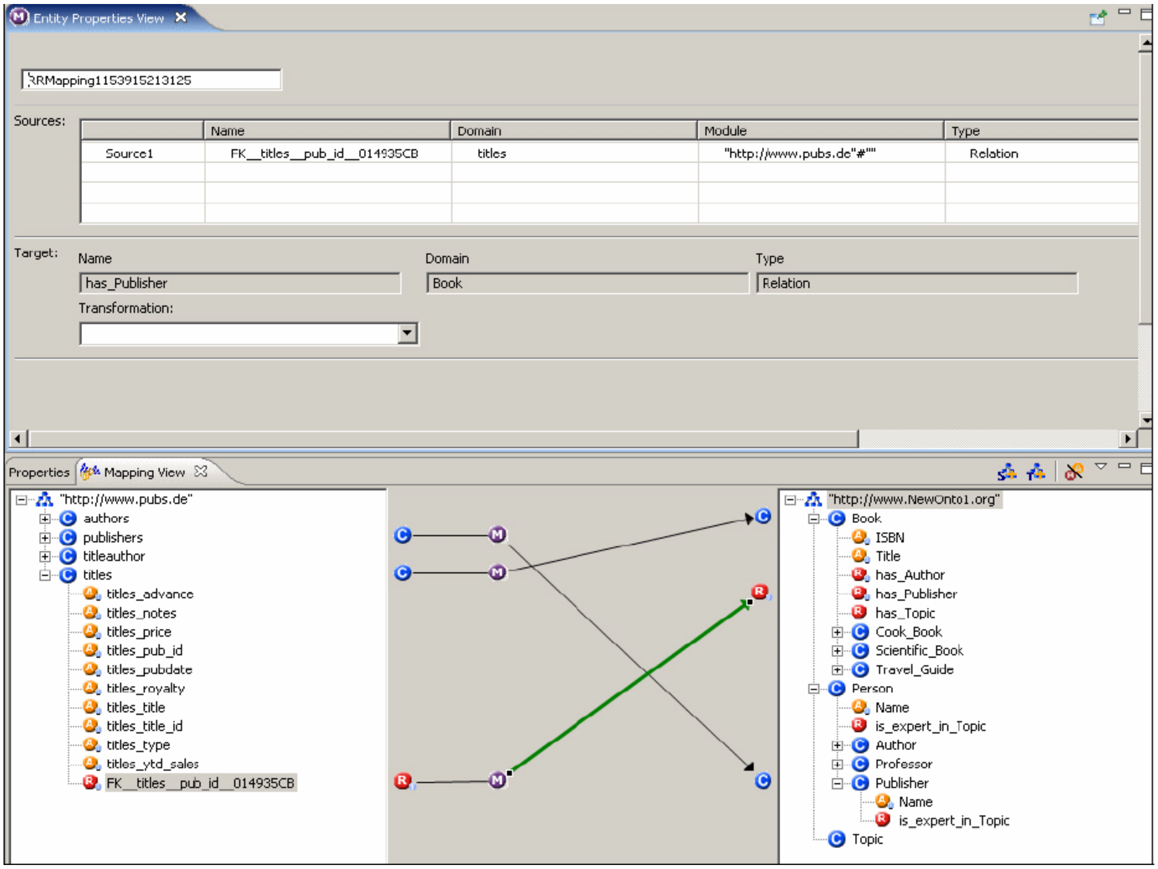

Figure 2.8. *Screenshot of NeOn toolkit mapping editor [NE008].*

databases, file systems, UML diagrams) into an ontology to be used for mapping.

Common to all of these tools is their support for OWL ontologies, which is the standard ontology language. They each use their own mapping formats, but the formats follow a similar description as previously discussed in Section 2.3. Also, each of the tools supports a semi-automatic process where the user works to validate automatically generated mapping correspondences. However, little user-based evaluation of these tools has taken place, and the few existing studies focus primarily on algorithm effectiveness without an explanation of the results. This is not surprising as no theory on how users define a mapping existed at that time. In this thesis, we begin the process of discovering this theory.

2.4.1 Mapping tool evaluation

As mentioned, current evaluation procedures for all of these tools have focused on the evaluation of the produced mappings in comparison to known mappings. PROMPT is the only tool where the tool authors performed a user evaluation experiment [NM02]. The experiment concentrated on evaluating the correspondence suggestions provided by the tool by having several users merge two ontologies. The researchers recorded: the number of steps, suggestions followed, suggestions that were not followed, and what the resulting ontologies looked like. Precision and recall are used to evaluate the quality of the suggestions: precision was the fraction of the tool's suggestions that the users followed and recall was the fraction of the operations performed by the users that were suggested by the tool. The experiment only involved four users, which was too small to draw any meaningful conclusions. The authors stated that, "[w]hat we really need is a larger-scale experiment that compares tools with similar sets of pragmatic criteria [NM02, p. 12]."

Lambrix and Edberg [LE03] performed a user evaluation of PROMPT and Chimaera for the specific use case of merging ontologies in bioinformatics. The user experiment involved eight users, four with computer science backgrounds and four with biology backgrounds. The participants were given a number of tasks to perform, a user manual on paper, and the software's help system for support. They were also instructed to "think aloud" and an evaluator took notes during the experiment. Afterwards, the users were asked to complete a questionnaire about their experience. The tools were evaluated with the same precision and recall measurements as used in the previously described PROMPT experiment [NM02], while the user interfaces were evaluated using the REAL (Relevance, Efficiency, Attitude, and Learnability) [Löw93] approach. Under both criteria, PROMPT outperformed Chimaera, however, the participants found learning how to merge ontologies in either tool was equally difficult. The participants found it particularly difficult to perform non-automated procedures in PROMPT, such as creating user-defined merges.

Although some researchers feel that more comprehensive experiments focused on how people actually perform mappings is key to productivity gains in the various related areas of schema matching [BM07, FS07b], other than these few examples, there has been very little research on this topic. Moreover, there is a lack of visual paradigms for ontology mapping. We feel that due to this, much of the ontology mapping research never leaves academic labs. As ontology usage continues to increase this problem must be addressed.

2.4.2 Mapping algorithms

There has been a variety of approaches used to automatically or semi-automatically perform ontology mappings. For example, Euzenat *et al.* discuss over 20 different algorithms and tools in [Ee04]. Very few of these approaches take into account the communication that must take place for the user to verify the produced mapping. Instead, they concentrate on the metrics for determining similarity between ontology terms.

One of the most widely used methods for computing similarities is heuristic techniques applied to the schema or ontological description. Heuristics are generally applied in two different ways. First, they are applied to the labels in the ontologies to compute lexical similarity, and second, they are applied to the structure of the ontology to measure structural similarity between terms. Chimera [MFRW00] and PROMPT [NM03] use lexical similarities to make suggestions to a user. They first execute an ontology alignment algorithm that attempts to find similar matches on concept names, prefixes, suffixes, or word roots. They then use the user's feedback about the suggestions to make further suggestions based on structural similarities.

Structural similarity is often partitioned into two classes: internal and external structure [Ee04]. Internal structural comparisons measure similarity between concept properties, such as cardinality, range, and symmetry. External structural comparisons attempt to find similarities between the ontologies by considering the ontology as a graph where edges are formed from the relationships described by the ontology (e.g., `is_a`). Most ontology mapping algorithms/tools apply a hybrid approach. For example, QOM [ES04a], uses a large number of heuristics for calculating label similarity (e.g., edit distance, substring matches, exact matches), internal structure similarity based on set similarities, and external structure similarity. All of these heuristics are combined using a weighted sum and normalized into a single metric.

Another, less widely used approach is the instance-based or instance-level approach [DDH03]. Here, concepts are compared based on their instances rather than their representation. An instance is an actual value of a concept, for example, an instance of a concept "Professor", would be an actual professor, such as Dr. Donald Knuth. Concept similarity can then be measured by comparing shared instances. Another way to measure the similarity for an instance-based approach is to apply machine learning techniques to build classifiers for concepts. The Glue system is an example of this; it builds learning classifiers for concepts and then evaluates the joint probability distribution of the assigned instances [Ee04].

The final mapping approach that is sometimes used is based on mapping ontologies to a standard data dictionary such as WordNet [wor] or UMLS (Unified Medical Language System) [UML]. With this technique, the data dictionary acts as a canonical form for every ontology that needs to be mapped. Each ontology can be compared to the data dictionary and the most similar term in the data dictionary becomes the canonical representation of the ontology term. Overlapping correspondences to the same canonical term from different ontologies indicate correspondences between those ontological terms. The advantage of this approach is that you are working with a known dictionary of terms, allowing researchers or developers to specifically tailor their algorithms for the terms within that dictionary. The disadvantage is that correspondences may be missed if a suitable canonical term does not exist in the data dictionary.

2.5 Summary

Ontology use is quickly growing. Ontologies provide a shared and common vocabulary for representing a domain of knowledge. Standards, such as OWL, have been proposed to the W3C for the development of ontologies, and thousands of known ontologies now exist. However, ontologies often describe similar domains and in order to support interoperability correspondences between these ontologies must be created.

Developing a mapping is a very difficult process and as a result has received a lot of attention in the research community. Most of the research has been on developing techniques for automatically discovering mappings. The relationship between users and the underlying mapping algorithms used by software tools is generally ignored. In ontology mapping, researchers tend to emphasize the algorithm component, however, it is important to consider the user's perspective in order to generate the best mapping. Supporting the user goes beyond simple user interface enhancements. For example, in ontology mapping, when algorithms report correspondences between concepts that are not obviously correct, understanding how and why the algorithm made this decision is important for the user so that they can properly validate or reject the correspondence. Thus, a user may actually perceive a less sophisticated algorithm as more useful than a technically more accurate algorithm if the algorithm lacks the ability to "explain" its results.

In a complex task, like ontology mapping, the relationship between the user and the tool has to be symbiotic. The user depends on the tool to help reduce the complexity of the task, while the tool relies and receives reinforcement from the user in order to guide the iterative nature of the underlying algorithms. In the next chapter, results from cognitive psychology on *categorization*, *human inference*, and *decision making* are presented. This builds on the discussion of some of the problems presented in section 2.3.2 on why mapping is difficult.

Part II

Theory building

Chapter 3

Human inference and ontology mapping

It is common knowledge that humans have short-term memory limitations [Mil], but there are other important limitations to human cognition. In this chapter we summarize relevant work from cognitive psychology on categorization, human inference, and decision making. We also summarize results from three behavioral studies that help illustrate the significance that categorical knowledge influences human inductive judgements. This discussion is important in order to understand what limitations and biases may influence users during an ontology mapping and comprehension task. We use this understanding to outline important implications for ontology mapping, addressing research objective **O1**. Parts of the literature review, experiments, and results were previously discussed in [YF08].

3.1 Human inference

Humans use inference during ontology mapping to make decisions about concept comparisons. An inference, in a way, is the extension of a property from concept *A* to another concept *B*. For example, knowing that plant *A* is poisonous, we may determine that plant *B* is poisonous based on observable shared properties. It has been suggested that this type of category-based inference simplifies the process required to experience all the unique events we witness in our daily lives [HB00]. Categorization limits the information we need to consider during inference [YM00]. Also, categories help provide us with simple "explanations" and "interpretations" of phenomenona we experience [Kei]. For instance, if someone is describing a building, and they label that building as a "house", then that categorization immediately allows us to make inferences about that building. These inferences are essentially predictions about the characteristics of that building. Since it is a house, we assume it has certain features common to other houses that we have had prior experience with.

Similar predictions or inferences take place during ontology exploration and mapping. Given a concept label, we use that label to represent a category of objects. Provided the objects across two category labels are highly similar, then a human may decide that those category labels, and hence the ontology concepts, represent the same thing.

This categorization process is fundamental to human inductive inference. We appear to carry out this process easily. However, category learning is a difficult task and there are costs associated with constructing an incorrect categorization [HB00]. Our goal in this research is to discover how to best support this kind of process during ontology mapping, but how to do this raises several important fundamental questions. How did we acquire the ability to make inferences? Are there systematic errors we tend to make while making inferences like this? By investigating and understanding these questions, we can discover what human factors are important for ontology mapping systems. We begin by presenting related work from cognitive psychology on *object permanence* and inference.

Object permanence refers to the fact that an object exists permanently whether we can see it or not, unless some external force modifies it [Mur02]. Children have quite a sophisticated understanding of this even at a young age. Brown [Bro57] showed that preschool-aged children used linguistic categories (i.e., count nouns, mass nouns, and verbs) to assign meaning. In the experiment, children were shown a picture and the picture was described using a meaningless word *zup*. Three different descriptions were used, where the word was present as a verb, count noun, and mass noun. For example, as a verb phrase, "This is *zupping*", as a count noun, "This is a *zup*", and as a mass noun, "This is some *zup*". The children were then shown three other pictures displaying motion, an object, and mass and were asked to select one picture as an example of the first. When the verb phrase was used, children tended to select the action picture. As a count noun, the object picture was selected, and finally, as a mass noun, the mass related picture was selected. The implication of this experiment was that children make use of a procedure known as *syntactic bootstrapping*, in which the syntax of a word provides initial cues for inferential projection [NS07].

Related to this, Gelman *et al.* showed that category labels lead to changed expectations, which can have both positive and negative effects [GH99]. The authors showed that categorical noun labels lead to inferential biases in four to five year old children. In their experiments, hypothetical characters were described as having certain characteristics and each character was described with a noun label condition or a verbal-predicate condition. For example, “She is a carrot-eater” as a noun condition, versus “She eats carrots whenever she can” as a verbal-predicate condition. Children were then asked a series of inference questions about the person described, e.g., “Did she eat carrots in the past?” and “Will she eat carrots when she is a grown up?”. The authors found that children associated the properties central to the identity of the character as more stable in the noun label condition. In separate experiments, Gelman *et al.*, found that children predicted objects grouped by a common label had shared properties/features even when the objects differed in appearance [GM86].

Similar results have been shown in studies involving adults. Markman tested whether a noun label supports the creation of more inferences and is considered more stable than an adjective label [Mar89]. In one experiment, participants were asked to list properties of categories, where the categories were either presented as nouns or adjectives, e.g., “an intellectual” versus “intellectual”. The participants were able to list more properties for the noun condition than the adjective condition. Also, Yamauchi *et al.* demonstrated that concrete noun labels create reasoning biases and that category labels are not just another feature on par with other category features [YM00]. Instead, the category label guides a person’s attention. Categorization appears to increase the perception of within-class similarity and between-class differences [TW] and this effect is exaggerated when grouped by noun labels.

3.2 Decision making

Human inference influences the types of decisions we make. In ontology mapping, one of the primary responsibilities of the user is to make decisions about potential correspondences that are computed by the automated procedure and to also manually create correspondences missed by the algorithm. In this section, we briefly review several decision making theories from cognitive psychology, and later, throughout the discussion of our experiments, relate these to observations from our studies.

Recently, in many areas of science, our understanding of problem solving and decision-making has had major gains. Research has demonstrated that people often solve problems by selective, heuristic search through large problem spaces and large databases [SDH+86]. Experts, such as chess masters, use these techniques to solve complex problems. They cannot analyze all possibilities from one chessboard state; they must prune their search space by using a variety of heuristics. Humans have successfully adapted heuristics of this type into computer chess playing systems. This type of decision-making process is known as the *heuristic-systematic persuasion model* [Syq].

People are able to reduce problems and complexity down to manageable sizes. The exact details of how people do this is not known. However, there has been a growing realization in the sciences that coping with complexity is central to human decision-making [SDH+86]. Besides heuristics, humans, especially experts, are able to access large amounts of information that is stored in memory when an appropriate cue exists to signal its relevance. For example, doctors diagnose patients based on symptoms as described by the patient or collected through tests. These symptoms act as cues, which trigger their knowledge about a possible cause. In ontology mapping, appropriate contextual cues about a concept's meaning must exist for the expert to validate concept correspondences.

The *elaboration likelihood model* [Syq] suggests that we make decisions in two ways. The first is when we are motivated about the problem and decision. In this case, we take a *central route* to decision-making. We do this by paying attention to the situation and thinking logically. On the other hand, if we are not motivated, then we take a *peripheral route* where our decision can be swayed based on surface characteristics of the problem.

Related to the decision-making process is the actual deciding that we do. There have been a variety of theories suggested in the literature. Below, we summarize a few of these that we believe relate to ontology mapping decisions.

Filter theory: suggests that we make decisions through a series of selection filters [KD62]. For example, a doctor may begin by asking a patient about their general symptoms and then continue to narrow the focus of the questions based on which possible diagnoses match the symptoms. In this case the symptoms act as a filter for possible choices.

Perceptual contrast effect: describes the effect that humans often make decisions by comparing and contrasting a decision item with a reference item.

Bounded rationality: was introduced to combat decision models that assumed humans to be fully rational [NS72]. It suggests that there are limits to human rationality, such as a limited capacity to understand everything, and we often make decisions with limited time.

Explanatory coherence: suggest that we build explanation hypotheses when we are trying to understand something [Tha89]. We prefer explanations that have greater breadth, are simple, and are plausible.

Multi-attribute choice: describes the decisions we make when comparing situations/objects with multiple attributes [EH81]. We tend to compare shared attributes or focus on the differences in order to come to a decision. For example, when deciding on which computer to purchase, we compare the shared features or attributes of the two machines as well as the differences.

Involvement: discusses how emotionally involved versus non-involved people make decisions [PC86]. Emotionally involved people want to make their own decisions and

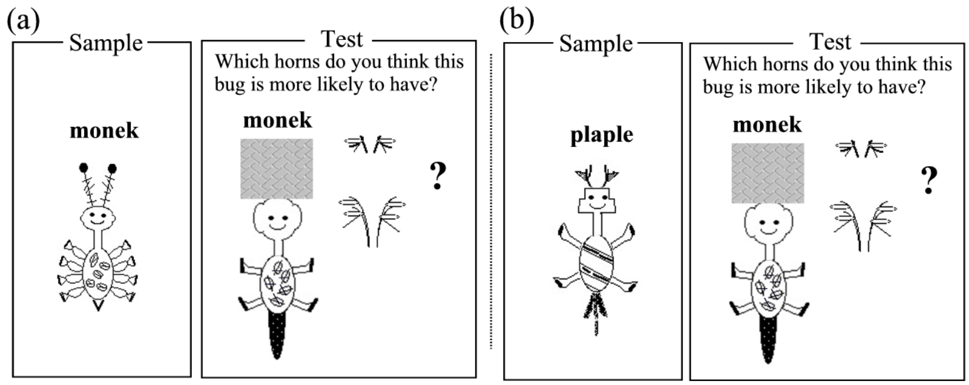

Figure 3.1. *Example of two stimuli [Yam07]. In (a) the insect labels match, while in (b) they do not. In both scenarios, the participant must predict what horns the Test insect has.*

also pay more attention to the details involved in the decision. Non-involved people do not want to put effort into their decisions and will happily let someone else tell them what to think. Emotionally involved people can become non-involved if they are overloaded with information as the quantity can overwhelm them.

All of these theories are important for understanding the cognitive processes involved in performing mapping tasks. By recognizing any of these decision theories in our experiments, we will potentially be able to introduce techniques to help support the decision making process. In the next section, we take a more in-depth look at human inferential judgement by presenting results from three behavioural studies. These studies investigate how humans make judgements based on ontological features, such as class labels and property values.

3.3 Behavioural studies

In the following sections we summarize results from three behavioural studies that were first presented in [Yam07, YF08]. These studies were conducted to investigate categorical knowledge in inductive reasoning, default strategies for inference, and the role of language during induction. They were specifically designed to test the types of inference and induc-

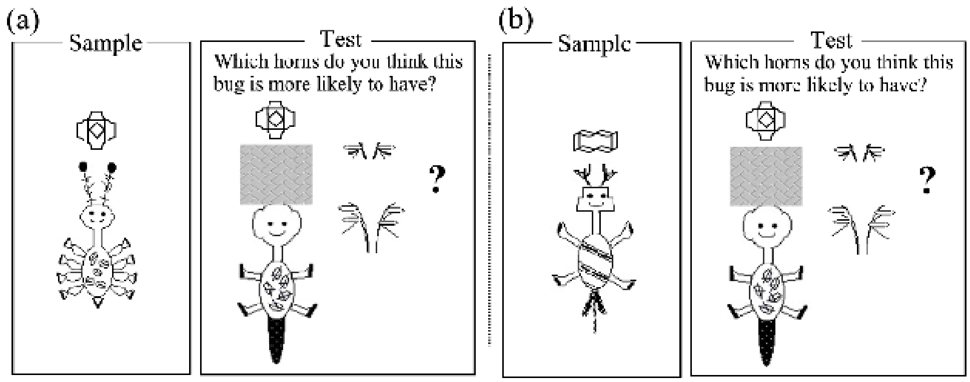

Figure 3.2. *Example of two stimuli with pictorial labels [Yam07]. In (a) the insect labels match, while in (b) they do not. In both scenarios, the participant must predict what horns the Test insect has.*

tion that is carried out during ontology creation, exploration, and mapping.

3.3.1 Study 1

The first study investigated the role of categorical knowledge in human inductive reasoning [Yam07, YF08]. Participants in the study were given pictures of cartoon insects, side by side, and were asked to predict the value of a hidden attribute of a test insect based on the sample insect. The value of the attribute could be either consistent with the sample insect or inconsistent with the sample. For example, in Figure 3.1(a), selecting the large horns for the test subject would be consistent with the sample insect, while selecting the small horns would be inconsistent. Arbitrary labels were assigned to each insect (e.g., "monek" and "plaple"). When the labels matched, it was assumed that the subjects would be more likely to choose the consistent feature value. The goal of the experiment was to test how this judgement would be influenced when the labels were assigned different ontological information, that is, class-label versus property possession.

In the class-label condition, the participants were instructed that the label represented the type of insect, while the property possession condition represented the shape of the wing the insects possessed. To test the extent to which textual labels influence judgement versus another means of representation, some participants received the same stimuli where class and property labels were represented by pictures rather than text (see Figure 3.2).

As was assumed, when the test and sample insects had the same labels, the participants were more likely to predict the insects as sharing similar properties and when the test and sample insect labels were mismatched, this consistency declined. However, interestingly, this effect was smaller when labels represented property information rather than class information and the pictorial labels appeared to be less important for inference than the textual labels. Overall, it appeared that participants relied more heavily on textual class labels for making feature comparisons than pictorial or property labels.

3.3.2 Study 2

In this experiment, the same cartoon insects from the first experiment were used [Yam07, YF08]. However, participants were told in the instructions that the class labels were determined randomly by a coin toss. This random condition was compared to meaningful property conditions, where labels represented properties such as a disease the insects carried or the name of the island the insects are found on. The goal was to demonstrate that class-labels are part of our default reasoning strategy. If this is true, then participants should still be more likely to predict similar property values for insects with the same class label or different values for different class labels when compared to the property label condition even though the class labels were randomly assigned.

The result indicated that this hypothesis was true. The impact of the random class labels was still higher than the property label condition. This implies that humans tend to rely on class labels somewhat automatically.

3.3.3 Study 3

In this third experiment, the role of language was investigated to determine the inductive potential of class labels [YF08]. This gives insight into why people tend to use class labels even when they are not meaningful.

The experiment compared the influence of categorical statements versus non-categorical statements. For example, "Linda is a feminist" is a categorical statement. It assigns the person "Linda" to the category "feminist". A non-categorical statement meaning essentially the same is "Linda believes in and supports feminism." In the experiment, participants were presented with premises such as these along with a conclusion, and had to estimate the strength of the conclusion based on the premise. Two examples are shown below.

Argument 1

Premise: Linda is a feminist.

Conclusion: Linda likes chicken fried rice.

Argument 2

Premise: Linda believes in and supports feminism.

Conclusion: Linda likes chicken fried rice.

There is no obvious link between the premise and conclusion in the examples shown. One would assume that the strength of the conclusions for both stimuli should be roughly equal. The only way to find strength to support the conclusion is by making up an explanation. The hypothesis in this study is that categorical statements increase the perceived likelihood that the conclusion follows the premise even when they have no obvious link.

The results supported this hypothesis. Participants in the categorical statement condition were far more likely to endorse the conclusion than those in the non-categorical statement condition. This appears to be related to the fact that categorical statements assist with processes such as justification and generalization, as was discussed above in the related work on human inference.

3.3.4 Conclusions

Based on the study results and the literature review, we arrived at the following conclusions:

- People overuse class-inclusion information to make an inference, even when the in-

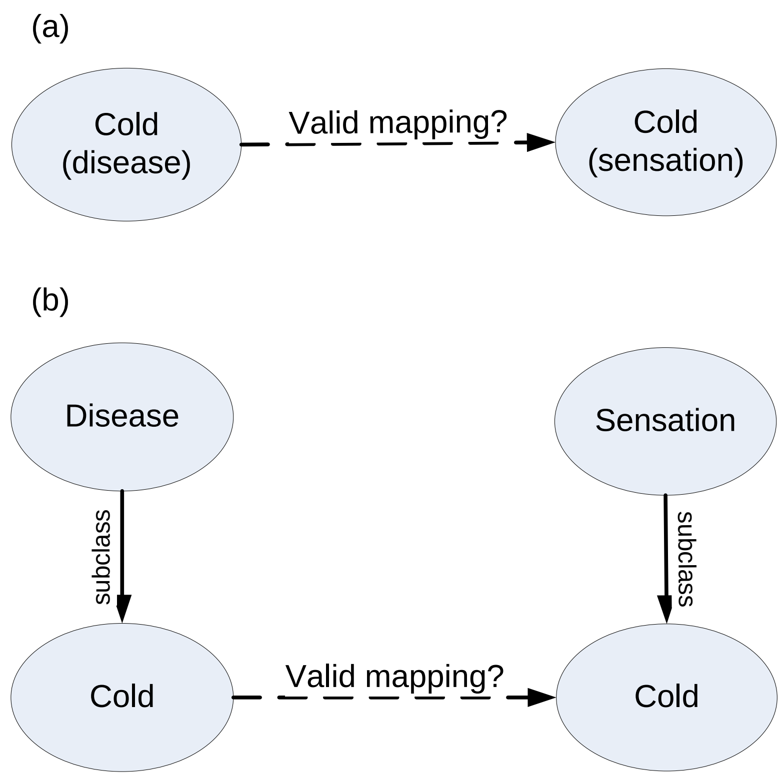

Figure 3.3. *Example of mapping scenario where context is import. In (a), the user must determine if "Cold" on the left should be mapped to "Cold" on the right. In (b), the parent concepts are shown. With the given context, the two terms should not be mapped even though they lexicographically match.*

formation is inadequate.

- The overreliance on class-inclusion information is partly reinforced by linguistic information (i.e., is-a statements).
- People tend to employ some default strategies to justify their judgements, and categorical statements encourage this tendency.

3.4 Implications for ontology mapping

In the following section we outline several important implications or human factors for ontology mapping based on the conclusions previously introduced. These implications help guide the development of the cognitive support framework introduced in Chapter 8.

People tend to overuse class-inclusion relations in making inferential judgements. Ontology mapping tools need to supplant this shortcoming.

Recognizing that we rely heavily on categorical statements for human inference has important implications for ontology mapping tool design. Many mapping tools provide output in some variation of the following form: $< uri1 >, < uri2 >, < similarity >$, where $uri1$ is a concept from the source ontology, $uri2$ is a concept from the target ontology, and $similarity$ is a numeric value of similarity computed automatically in the range [0, 1] [ES05]. However, the taxonomic structure of an ontology as well as the properties of a concept specify the context of a term. The term alone or the URI reference alone is not always enough for a human to determine whether the concepts actually mean the same thing (see Fig. 3.3).

The results from Studies 1-3 imply that users are likely to default to comparing the results of the automatic procedure based on the categorical information of the URI references, rather than actually inspecting the context of the term usage. It is important that mapping tools make this context readily available, make its inspection part of the mapping process, and also highlight contextual similarities and differences between the concept. Otherwise users are likely to make mistakes.

Properties are not equal in their importance. Some properties are more important in determining class membership.

The expressiveness of properties is critical in ontology mapping for determining correspondences between concepts. Mapping algorithms such as NOM and QOM [ES04a] compare the set similarity between concept properties to help determine concept equality. All properties of the concepts being compared are treated equally. However, for many concept comparisons, treating all properties the same will introduce noise and lower the real actual similarity.

For example, consider comparing concepts “Watch” and “WristWatch”. “Watch” has properties “TellsTime” and “HasStrap” while “WristWatch” has properties “CanTellTime”, “Strap”, and “Color”. Intuitively, the properties “TellsTime” and “CanTellTime” are considerably more important to the human concept of watch than a property like “Color”, and the fact that concept “Watch” does not have the “Color” property should not influence its concept similarity to the same degree as being able to tell time.

Mapping algorithms as well as structural representations of a concept in a mapping system user interface could potentially use a metric such as TF-IDF [Jon72] that takes into account how rare a property is to help determine how distinguishing that particular property is. Properties like “name” are common and they are not distinguishing characteristics or features of a concept. Potentially, a unique concept property is more important for algorithmic comparison and inferential judgement by a user.

Distinguishing similarities and differences in properties when making concept comparisons could facilitate human decision making. As mentioned, the multi-attribute choice theory states that humans compare both shared and different properties for object comparison. Visual tools for concept comparison in mapping systems could facilitate this process.

3.5 Discussion

We need to harness the relationship between humans and machines to produce better ontology mapping systems. Coradeschi and Saffioti [CS06] suggest that the future of intelligent robotics lies not in the development of fully automated robots, but partially automated robots. Some tasks, such as classification and pattern recognition, are very difficult. Robots need help from humans, while robots can help humans with tedious and repetitive tasks. Similarly, in ontology mapping, humans have access to vast amounts of background knowledge, which can be used to help make inductive judgements about potential correspondences. However, humans can make mistakes by overrelying on concept labels. Mapping systems can help mitigate this by drawing the user’s attention to the properties of a concept during a comparison task, in particular, the shared and unshared properties.

Recently, in mapping related research, this symbiosis has been gaining more prominence. Both Optima [KD08] and AlViz [LS06] use information visualization techniques to help guide the user through this process. Also, the *International Workshop on Ontology Alignment and Visualization* [1] is scheduled for a second year, which is the first workshop focused on issues of usability in mapping. Moreover, Shvaiko *et al.* [SE08] discuss ten challenges for ontology matching, three of which directly relate to the user: *user involvement, explanation of matching results,* and *social and collaborative ontology matching*.

The Potluck [HMK08] system incorporates a user-centric metaphor for the transformation or semi-automatic alignment of ontologies. Users work directly with the raw instance data that is encoded by Exhibit powered web pages. Potluck supports a drag drop interface for cleaning the data and for aligning different fields across multiple schemas. Visualizations of the data can be constructed in place, giving the user immediate feedback about the alignment they are constructing.

There are also a few projects that are attempting to address problems of data heterogeneity by making use of communities of users. The notion of Community-Driven Ontology Mapping is discussed by Zhdanova *et al.* [Zhd05], and a similar practise is used in BioPortal [NGM08]. Although these approaches do not specifically address usability, they are trying to support users with application specific needs, as well as use the power of a community effort to develop mappings.

3.6 Summary

In this chapter we presented a literature review of categorization, human inference, and decision making. We also summarized three different behavioral studies related to human inference and categorization. Categorization is fundamental to human inference and we use it to help us explain and understand the world. However, categorization is intertwined with our language, and how we "label" categories influences the properties we associate

[1] http://www.ifs.tuwien.ac.at/~mlanzenberger/OnAV09/

with members of that category. This has important implications for ontology mapping tool design, namely that we must realize that users will tend to base their decisions primarily on concept labels when making mapping decisions. On the algorithm side, researchers must realize that not all properties are of equal importance. Finally, to mitigate these implications, we must take a human-machine symbiotic approach to ontology mapping tool design. In Chapters 5, 6, and 7, these implications help guide our experimental analysis and eventually our design principles.

In the next chapter, we discuss relevant theories of *cognitive support*. These theories, along with the theories outlined in this chapter, play an important role in determining how to best support the human-machine symbiosis necessary for semi-automatic ontology mapping systems.

Chapter 4

Cognitive support

In this chapter, we begin by defining cognitive support. Following this, we discuss some of the implications of automation in software and then discuss relevant theories of cognitive support from software and ontology engineering. We highlight three theories of cognitive support introduced by Walenstein [Wal02c]: *redistribution*, *perceptual substitution*, and *ends-means structure reification*. These theories help guide our study analysis and the construction of our cognitive support framework introduced in Chapter 8. Also, in Chapter 10, we use these three theories to help guide our tool evaluation.

4.1 What is cognitive support?

In software engineering, there is usually some emphasis placed on the usability of tools, but only recently has our understanding of how to make tools usable moved towards taking a more formal approach. Researchers and tool developers alike often describe usability in terms of providing *cognitive support* or reducing *cognitive load*; however, in order to move beyond *ad hoc* strategies for support, we need to really understand the meaning of these terms. Simply put, cognitive support refers to the assistance, or aid, that tools provide to humans in their thinking and problem solving [Wal03]. It is about the introduction of external artifacts to improve cognitive processes [Wal02a].

Humans often rely on external artifacts to support their cognition. For example, a sticky note can be used as an external memory source - a reminder about a task we need to complete. In software tools, software artifacts (e.g. menus, search, term completion) can be introduced to support the human user's cognition.

The goal of cognitive support within a software system is to offload some of the user's cognitive processes involved in performing a task to the software. By doing this, we reduce the number of items that user must internally track and process, allowing them to apply their attention and memory resources to other parts of the task and thus make the task easier to perform. This goal relates to *Cognitive Load Theory*, first defined by Sweller [Swe88]. The theory suggests that "instructional techniques that require students to engage in activities that are not directed at schema acquisition and automation, frequently assume a processing capacity greater than our limits and so are likely to be defective" [Swe94, p. 299] or more simply, "optimum learning occurs in humans when the load on working memory is kept to a minimum to best facilitate the changes in long term memory" [Cog].

Psychologists have found that the more things a person has to learn over a short period of time, the more difficult it is to process information in working memory [Cog]. As an example of this, it is generally much easier for students to learn how to perform long division by first learning how to add, subtract, and multiply versus learning long division first. The reason for this is because addition, subtraction and multiplication are all sub-components of learning long division. Hopefully, by the time long division is introduced to the student, performing the sub-components are second nature, which leaves their working memory free to concentrate and process the new components of the task.

4.2 Implications of automation

In research and industry, there is often a tendency to try to support users by automating tasks. However, some tasks are too difficult to fully automate, and it is up to the user to deal with most of the task complexity. Automation can sometimes introduce new complexity or frustration for users (e.g., the seemingly endless menu options that must be navigated when dealing with automated phone systems). Brainbridge observed that automation often provides the least assistance when we need it most, as generally we can only automate rudimentary tasks [Bra83].

This relates to the previously discussed user evaluation of PROMPT and Chimaera discussed in Chapter 2. The participants noted that performing non-automated procedures with PROMPT was difficult. This is true with many mapping tools, as the tools can only automatically discover the simple correspondences. It is then completely left to the user to manually create the rest of the correspondences, which most tools lack adequate support for.

Billings [Bil91] and Norman [Nor93] studied automation in airplanes and its effect on pilots. They noted that automation can introduce its own problems and complexities. Billings found that automation increases the perceptual and cognitive demands of the pilot by increasing the overhead involved in actually operating the automation. Endsley *et al.* discussed situation awareness with automation and noted that traditional automation design does not consider the cognitive needs of the users [End95].

In software development applications, automation often takes the form of tools like compilers, but it is ultimately up to the human programmer to resolve complex debugging issues and to design and implement algorithms. However, the programmer can be supported by introducing various cognitive aids. Popular IDEs, such as Eclipse and Visual Studio, provide cognitive support for the user. An example of this is code completion. As a programmer types in code, the IDE produces a list of candidate possibilities from which the programmer can select to complete the command. Without this support, the programmer would have to either recall the full command or look the information up in an API. Code completion changes the nature of the task from a recall problem to a more easily performed recognition task.

4.3 Cognitive support in software and ontology engineering

In this section we discuss previous cognitive support work from software and ontology engineering. We begin by presenting work from Robbins and Walenstein, both of which investigated cognitive support for software engineering tools. We then briefly describe Walenstein's work on measuring the cognitive support provided by a tool and Ernst's work on cognitive support for ontology navigation.

Robbins addressed cognitive support for software engineering design tools like UML modeling systems. He drew on existing theories from cognitive science about design decision-making in order to address the cognitive needs of software designers. His goal was to develop a better software architecture design environment based on identifying cognitive theories that describe people's behavior in design situations [RR96]. Robbins evaluated his software design tool by a combination of heuristic evaluation of the cognitive features using cognitive walkthroughs and a series of user studies. His controlled lab studies focused on specific features of the tool, for example, his model alignment component. In this experiment, he demonstrated that on average, the mouse was moved 86% further using standard tools than his tool. He also evaluated his tool by observing students using it in a classroom and through Internet feedback.

Walenstein developed a framework that analysts could use in order to think in terms of cognitive support. He called this framework "RODS", which consists of four cognitive support classes: task Reduction, algorithmic Optimization, Distribution, and Specialization. Each of these classes specify computational principles that specifically address issues relating to human cognitive needs. These four classes are discussed in detail in [Wal02a, Wal02b].

Robbins's and Walenstein's research approaches were similar. They identified a need for cognitive support in software engineering and searched for relevant related research from other disciplines such as psychology. The related theories and models were then adapted to suit their domain. While Robbins used his set of adapted theories to help identify important cognitive support features for a software design tool, Walenstein proposed his own theoretical framework for cognitive support in the domain of reverse engineering.

Walenstein was also active in developing techniques for quantitatively measuring cognitive support. He developed his techniques and the RODS framework by building on the theory of *distributed cognition* [Hut95]. The theory argues that cognition is not a local process restricted solely to the human mind, but one that is "distributed" by placing memories, facts, or knowledge on the objects, individuals, and tools in our environment [Nor94, wik06]. Walenstein states that "if cognition is computation, and if cognition can be distributed between humans and artifacts, then artifacts support cognition by re-engineering the computational processes involved" [Wal03, p. 187]. In order to measure cognitive support, the computational benefit that a tool provides by re-engineering the cognitive processes must be observed and measured.

Walenstein [Wal03] described an experiment where he measured the cognitive support of the debugging tool in Visual Café. He performed this experiment by first analyzing how a human performs a program repair task with and without the tool. This step was performed to determine how the tool re-engineers the cognition involved with repairing software compilation errors. It was determined that fixing software errors is related to goal and plan following and a simple model representing the cognitive processes involved in this strategy was constructed for a human with and without the tool.

The next step was to use this analysis to devise a coding scheme that could be used for observing the workings of cognitive support. The coding scheme was designed to capture both the programmer's internal cognition as well as the external cognition provided by the tool. As an example, one code *GI(g)* coded for "add goal *g* to agenda (from internal)" while another code *GE(g)* coded for "add goal *g* to agenda (from external)". Finally, to quantitatively measure the cognitive support provided by the tool, a user study was conducted. Participants were observed for approximately 40 minutes, during which they carried out their regular development duties. The participants were instructed to generate verbal reports while they worked and the sessions were video recorded. Using the data collected, the coding scheme was applied and frequency counts for each code was calculated. This calculation provided the quantitative data necessary for the analysis. For example, Walenstein found that 15 out of the 31 coded events related to generating or manipulating plans and goals [Wal03].

Cognitive support has also been studied in the domain of ontology engineering. Ernst *et al.* [ESA05] studied cognitive support in this domain and identified three main task areas requiring support: navigation, modeling, and verification. In ontology mapping, navigation and verification are also important issues. Mapping users must navigate the ontologies as well as potential mapping correspondences. They must also be able to verify that the correspondences they make are correct or gather evidence from the structure of the ontologies to make an informed decision about a mapping.

4.3.1 Theories of cognitive support

In this thesis, we build on three cognitive support theories discussed by Walenstein: *redistribution*, *perceptual substitution*, and *ends-means structure reification* [Wal02c] as well as the mapping implications identified in Chapter 3. A cognitive support theory, as defined by Walenstein, is "a generalized statement about how and why some abstract class of artifacts (and their uses) manage to make cognition better" [Wal02c, p. 75]. Redistribution is the idea that cognitive resources and processing can be moved outside one's "head" and into the outside world. In terms of software tools, the artifacts or features of the tool, such as menus, search, and term completion, are resources the user can rely on to help them complete a task. Perceptual substitution is the transformation of a task that allows for fast cognitive operations to be substituted for slower cognitive operations. For example, a pie chart comparing market share between three different companies allows for fast visual processing, whereas a table displaying the same information requires more mental work to compare and contrast the percentages. Finally, ends-mean reification is the process of representing a set of means to allow someone to progress towards a goal. For example, programming environments like Eclipse and Visual Studio provide a list of compile errors. This list gives the programmer a set of specific tasks to complete to meet the goal of compiling their source code.

Walenstein had previously used the theories to rationalize tool design for program comprehension. He proposed a design heuristic based on the theories, where the heuristic states to 1) maximize redistribution, 2) substitute perceptual operators wherever possible, and 3) reify the ends-means mapping structure. We chose to build on this work as we recognized that program comprehension and reverse engineering share similarities with ontology engineering and mapping. Software is typically organized into a hierarchical class structure similar to that of an ontology. Also, Walenstein's work in part built on Hutchin's work on distributed cognition. We believe this theory fits well with the human-machine symbiosis and mapping implications discussed in Chapter 3.

In Chapters 5, 6, and 7, we use these cognitive support theories to guide the discovery of opportunities for cognitive support during the analysis of our experiments. We also use background from the discussed theories of categorization, human inference, and decision making to help us understand the processes and limitations of ontology mapping users.

4.4 Summary

In this chapter we defined cognitive support as the aid a tool provides during thinking and problem-solving. We discussed the implications of full automation and how automation can have unforeseen consequences. We briefly outlined previous work on cognitive support theories in software and ontology engineering. We primarily focused on work by Walenstein. His theories of support, based on distributed cognition, mesh well with the human-machine symbiosis discussed in the previous chapter. Later, in Chapter 10, we use the theories of redistribution, perceptual substitution, and ends-means structure reification to help guide our qualitative analysis during our tool evaluation. These theories are key to helping explain why users were able to perform more efficiently and effectively using our tool. In the next chapter we present the first of three exploratory studies that focus on uncovering the necessary elements of cognitive support needed in semi-automatic ontology mapping systems.

Chapter 5

Observational case study

In this chapter we present the first of three studies. The goal of these studies is to address research objectives **O2, O3** and **O4** (see Chapter 1 Section 1.2). These objectives are exploratory and as a result, each of our studies are exploratory in nature. We rely primarily on qualitative research methods for data collection and analysis. The study presented in this chapter was previously presented in [FS07a, FS07b].

5.1 Study design

In this study, we observed two teams of two users performing mappings with two different ontology mapping tools. The study design reflects our objective to understand and investigate how mapping users perform mappings between different data representations, and what the opportunities for cognitive support are. Specifically, we are interested in investigating the following research questions:

Q1: How do users make decisions during the mapping process?

Q2: Which parts of this task are particularly difficult?

Q3: Which parts of the task do existing tools support well?

Q4: Which parts do they not support well?

Each of these questions address areas of the mapping process that are potential opportunities for improved cognitive support. For example, we need to understand the decision making process in order to introduce cognitive aids that will support it.

In the study we used two different tools, COMA++ and PROMPT. These tools, introduced in Chapter 2, were selected for several reasons. First, they both support user-interaction and a graphical user-interface. However, both tools support this user-interaction slightly differently. COMA++ computes a full mapping between the ontologies and then the user interacts with the ontology trees to remove invalid correspondences and create correspondences missed by the algorithm. On the other hand, PROMPT produces a list of candidate correspondences that the user verifies by completing the suggested correspondence or removing the operation. This feedback is used by PROMPT to make further suggestions. Moreover, the user-interfaces for both tools are distinctly different. This is important as it allows us to investigate what type of interface better supports a user's mental model. Finally, both tools have support for OWL ontologies, meaning that the same ontologies can be used by both tools during the experiment.

Two university related ontologies were selected for the experiment, one developed at the University of Maryland (UMD) and the other from Carnegie Mellon University (CMU). These ontologies were selected because they represent real overlapping ontologies developed by two individual groups. Also, the ontologies cover a domain that should be familiar to all participants and are small enough (UMD approximately 135 concepts, CMU approximately 54 concepts) to be explored during the short duration of the experiment. It is important to note that these ontologies are relatively small compared to many existing ontologies that contain thousands of concepts and relationships.

5.1.1 Research approach

We used a qualitative research approach to address the posed exploratory research questions. We wanted to understand how users make decisions and also what cognitive aids could be introduced to support users during this process. Note that we were not interested in comparing the functionality of the mapping tools used, but rather how each tool supports a user's cognition.

5.1.2 Participants

Four participants, $P1$, $P2$, $P3$, and $P4$, were involved in the study. $P1$ and $P2$ were placed in the first team, $T1$, while $P3$ and $P4$ were on the second team, $T2$. Recruitment targeted users with significant computer experience (at least five years) and those comfortable with English. These two characteristics were important as the mapping tools were new to the participants. Thus, they had to be comfortable enough with computers to learn the basic functionality of the tools during a short training session. Also, the participants were mapping English ontologies. As a result, it was important to target participants that were comfortable with English. Recruitment took place by advertising in the Computer Science building at the University of Victoria (see Appendix A), and informed consent was given (see Appendix B).

$P1$ (male) is a Masters student in Computer Science and also a part-time programmer. He has been using computers for approximately 27 years and spends up to 60 hours a week working with a computer, primarily for programming (using Eclipse). Prior to the study, he knew what an ontology was, but had not worked with any ontology tools before. Also, he had not performed any data mappings prior to the study. He has a background in Human Computer Interaction (HCI) and pair programming.

$P2$ (male) is a PhD student in Computer Science and has been working with computers for over 16 years. He spends 60 or more hours working with computers a week, primarily for programming (using Eclipse). Prior to the study, he was familiar with what an ontology was and had worked with ontology tools before. He had not performed ontology mappings before, but had experience mapping XML Schemas and other models. He also has a background in HCI and pair programming.

$P3$ (female) is a Masters student in Ocean Science and has been working with computers for approximately 8 years. She spends up to 45 hours a week working with a computer, primarily for programming (in Matlab). Prior to the study, she was familiar with the term ontology, but had not worked with ontologies or ontology tools. She also had no prior experience performing data mappings.

$P4$ (male) has a Bachelor of Computer Science and Physics degree and currently works as a fulltime programmer. He has been working with computers for around 15 years and spends up to 60 hours a week using a computer, primarily for programming. He has experience working with ontologies as well as ontology tools, but no prior experience performing data mappings. He also has had experience working with a wide range of applications, such as Eclipse, Visual Studio, and multiple ontology editing tools. Finally, he has experience with pair programming.

5.1.3 Data collection

We administered a pre-study questionnaire in order to collect information pertaining to the participants' overall computer experience, ontology experience, and data mapping experience (see Appendix C). After the questionnaire was completed, a brief training period took place where the basic functionality of PROMPT and COMA++ was explained. Once all questions relating to ontologies and mapping were answered, the study began. Data for each session was collected using video and audio recordings. Also, the mapping produced by each team were saved for analysis. Each team was given approximately 15 minutes to perform mappings with each tool (30 minutes total). The use of the tools were alternated for each team, that is, one team would experience PROMPT first and then COMA++, while the next team would use COMA++ first and then PROMPT. During this part of the study, teams were instructed to "think aloud" [ES80] or discuss their decisions with the other team member. We motivated the problem of mapping the ontologies by instructing the participants to imagine that they were hired by the two universities, and that afterwards they may need to justify their mapping decisions to the people that hired them.

After the mapping section of the study was complete, a post-experiment interview was conducted with the participants. The questions in the interview focused on the teams' experience with the tools and with performing the mappings. The interview was designed to get the participants to further explain how they made decisions, how the tools supported their cognition, and how the tools could be improved to further support this process.

5.1.4 Analysis

The study was exploratory and the goal was to help gather requirements for cognitive support. As a result, we used a qualitative analysis approach. We constructed a story about each user based on the pre-questionnaire and other study results. For each team, we gathered the available data and performed a preliminary exploratory analysis [Cre03]. This involved organizing the data for analysis, watching the study sessions again and re-reading the submitted pre-questionnaires, and finally determining a coding process. Following this, the saved ontology mapping were coded/categorized. Ontology mapping algorithms generally combine heuristic measurements based on three general criteria: syntactic similarity between concept terms, semantic similarity between concept terms, and finally structural similarity [Ee04, SE05]. We coded the generated mapping according to these broad data divergence categories.

Also, the video recordings were annotated specifically when correspondences were created or validated by the users. This was done to evaluate the codings of the saved mapping with what the participants discussed during the actual mapping.

The video recording from the mapping section of the study along with the post-experiment interview were further analyzed for emergent themes across all participants. These emergent themes begin to form the basis for a set of cognitive support requirements for mapping tools.

5.2 Results

The results from this coding are displayed in Table 5.1 as well as in Fig. 5.1 and Fig. 5.2. The "Name Matches" are split between exact name matches and name matches with spelling variations. As we see, the name matches dominate the produced mapping. This was not particularly surprising. As was shown in Chapter 3, similarity judgement is largely dominated by categorical noun labels and to some degree this is a default strategy for human inference. Objects with the same label are predicted as belonging to the same class

Table 5.1. *Shows the coding of mapping data from both teams with both tools.*

Team	Tool	Name Matches		Synonym Matches	Structural Matches	Total
		Exact	**Spelling Diff.**			
1	PROMPT	10	2	1	0	13
	COMA++	10	3	6	1	20
2	PROMPT	17	3	3	0	23
	COMA++	8	2	6	1	17

even when their appearances vary [GM86]. Moreover, since the users only had 15 minutes with each tool, relying on this default strategy of inference makes these correspondences the easiest mappings to validate.

As mentioned, we also annotated the video files when the participants validated a correspondence and analyzed the participant's conversations in order to provide context for each correspondence. By synchronizing the conversations of the "think aloud" study with the mapping files, we see from the conversations that although most of the correspondences were exact name matches, the participants heavily relied on both the internal and external structure of the ontologies as confirmation that a correspondence was correct. Thus the name was used only as a hint that the match was probable. Moreover, $T1$ even rejected matches between "Email" and "EMail" as well as "Student" and "Students" because the structure of the concepts were different. The difference in structure led them to believe that although they shared the same concept label, they were in fact different concepts.

Using the coding (see Section 5.1.4), we also analyzed the overlap between the mapping produced by each tool for each team. With $T1$, 10 of the correspondences produced between PROMPT and COMA++ coincided, while with $T2$, 9 of the correspondences coincided. There was a small number of correspondence divergences between the tools. For example, one term was mapped differently in PROMPT than in COMA++ by $T1$, while $T2$ mapped two terms differently. Finally, we also analyzed the mapping overlap between the two teams. Twelve of the correspondences produced using PROMPT were the same

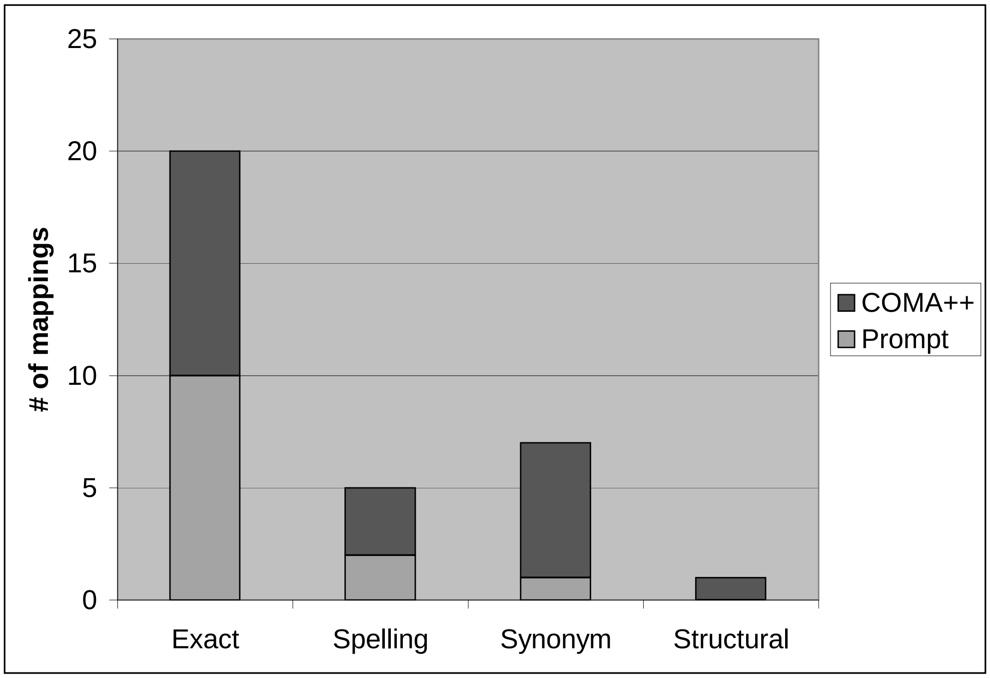

Figure 5.1. *Bar chart representation of $T1$'s coding results from Table 5.1.*

between $T1$ and $T2$, while 13 of the correspondences produced with COMA++ were the same.

From the coding, we also see that there was not a large difference in the number of produced correspondences between the tools. However, the "think aloud" study and interview revealed a large difference in the users' satisfaction with the tools. $T1$ felt that by far, PROMPT was the more useful tool. They had a lot of difficulty making sense of the mapping correspondence lines drawn in COMA++ and $T1$ started to ignore the mapping suggestions after using the tool for 7 minutes. Also, after this time, their productivity greatly improved. They started to rely on remembering what they had mapped before and also their knowledge of the ontology's terms. The participants also highlighted context switching issues

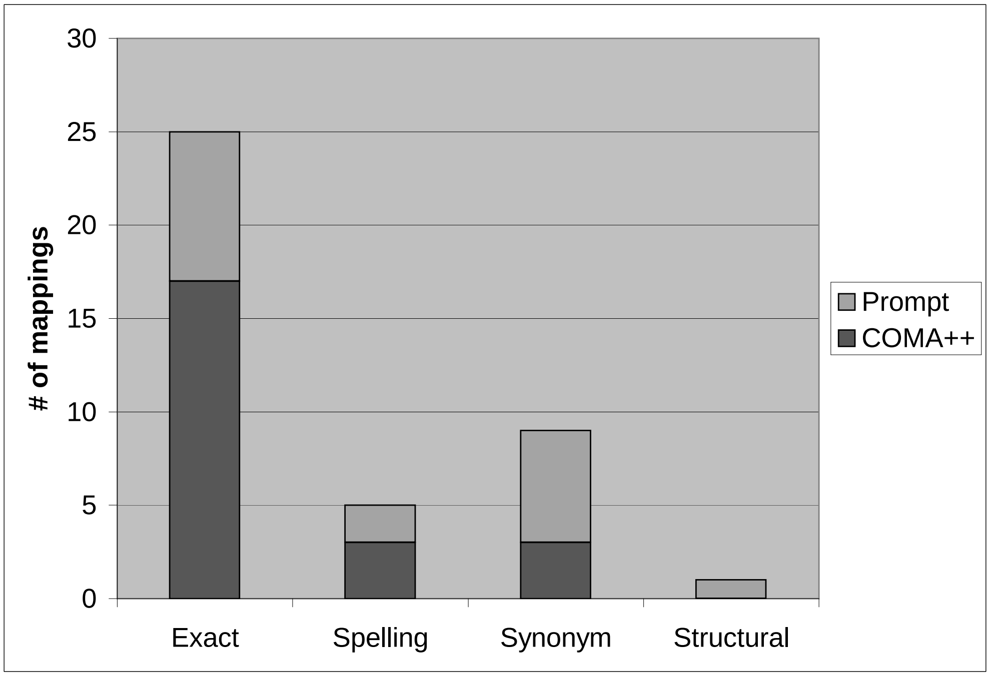

Figure 5.2. *Bar chart representation of $T2$'s coding results from Table 5.1.*

with COMA++. They found it difficult to tell what had been mapped and what was left to be verified or mapped. $P2$ even stated, *"How do we know when we're done?"*

$T1$ also stated during the interview that they felt two people were necessary to use COMA++ effectively because it forced them to remember so much information; where they were in the ontologies, what had been mapped, etc. They even mapped one term twice, first correctly, and then later incorrectly. The teams also tended to come back to terms they had already mapped, having forgotten that they had already inspected them.

Conversely, $T2$ primarily felt that COMA++ was the more effective tool. $P3$ stated, *"COMA++ was easy, was straight-forward, was obvious. The Protégé [*PROMPT*] tool was irritatingly complex."* $P4$ agreed that PROMPT had a complex interface, but he did not feel that either tool was necessarily better. He stated that COMA++ was simpler, but difficult to use when there were a lot of candidate correspondence lines. He did however feel more confident about the mapping he produced using COMA++. PROMPT gave more information for validating a correspondence, but that also complicated the process.

5.3 Findings

From the analysis of the data collected during the study, we identified a number of user-related issues. These issues or themes begin the process of forming a set of cognitive support requirements for human-guided mapping tools. It is important to note that we are not attempting to compare the tool's functionality. We are interested in investigating how each tool supports the user's process.

5.3.1 Decision making process

From our analysis, we observed that all participants followed a similar decision making process when judging potential correspondences. Consistent with the studies discussed in Chapter 3, they relied on concept name similarity from either the suggested candidate correspondences or the ontology trees as an indicator of a possible alignment. Next, they used both the internal and external structure of the concepts for validation. If the concepts had similar structure (i.e., context), they felt confident that the correspondence was valid. Reliance on the category structure (i.e., superclasses and subclasses) is also consistent with the previously discussed human inference experiments.

$T2$ also highlighted that they relied on their domain knowledge of how a university functions to make decisions. These observations directly correspond to some of the decision making theories previously discussed (Chapter 3). Exact matches allow the users to quickly filter the mapping suggestions, as in filter theory. Also, users rely on the internal structure of the terms to compare shared and unshared attributes to infer intended meaning. Domain expertise is used (as in the heuristic-systematic persuasion model) to search for appropriate correspondences and also contributes to confirmation bias when inspecting a correspondence.

5.3.2 Search and filter

The concept of being able to search and filter correspondences and ontology data came up several times during both the mapping session and interviews. PROMPT supports searching, but the functionality did not work as the participants expected. $T1$ mentioned searching repeatedly, especially while using COMA++, which does not have any facilities for searching the ontologies.

5.3.3 Navigation

In PROMPT, both teams relied primarily on the list of candidate correspondences for navigation, while in COMA++, the teams relied on the tree structure of the ontologies. With COMA++, since the user's only navigational device is the ontology trees, often when participants found a correspondence between two concepts, they were able to perform several mapping operations quickly. We believe this is because once they found a correspondence that they were sure about, they were able to use the ontology trees to infer other correspondences of parent and child concepts. However, in PROMPT, they primarily focused on the candidate list, and mostly ignored the ontology trees. Due to this difference in navigation strategy, we believe that COMA++ may better facilitate learning of the ontologies since the user must browse the trees to perform mapping operations.

5.3.4 Difficult mappings

As mentioned, most of the performed mapping operations were perceived by the participants as simple or "easy" correspondences. However, during the study sessions, both teams were forced to ignore some potential correspondences when they could not determine if the correspondence was correct or could not agree on a decision. This is an interesting result, as both tools do not support a mechanism for returning to a decision point. It is left to the user to remember to come back and inspect a correspondence that they initially ignore.

5.3.5 Mapping progress

Both teams emphasized the need for the tools to support a measurement of progress or a way to determine what has been mapped and what is left to map. $T1$ stated that PROMPT supported this better, as it places an “m” icon beside mapped concepts. However, they found it difficult to get a sense for how much they had accomplished, as well as understand how much is left to complete. $T2$ also had a similar experience, and felt that COMA++ needed to visually show the difference between an unverified mapping corresponce line and a verified correspondence.

5.3.6 Trusting the automation

The teams were often confused when a correspondence was suggested between two concepts that were obviously different. For example, PROMPT suggested that “Meeting” be mapped to “Thing”, stating that the names were similar. PROMPT calculates these names as being similar because both terms share the “ing” suffix; however, this similarity is not obvious to the users and only confuses them. This confusion leads to trust issues between the user and the automated part of the mapping tool. As stated before, $T1$ had a difficult time understanding the correspondences suggested by COMA++ and eventually ignored them completely, relying solely on their own exploration of the ontologies.

The teams both liked that PROMPT supplied a reason for suggesting a correspondence, although sometimes this reason led to confusion and indecision. $T2$ did not feel the confidence value provided by COMA++ (a number between 0 and 1) was particularly useful.

This finding is consistent with one of the ten challenges of ontology matching outlined by Shaviko *et al.* [SE08]. The authors explain that it is important to represent matching explanations in a clear and simple way in order to facilitate the user decision making process.

5.4 Limitations

Although we feel our study yielded valuable insight into how users interact and work with ontology mapping tools, since our study was limited to four participants, we cannot easily generalize our user-related issues/themes across all users. The study consisted of performing mapping operations for only 30 minutes (15 minutes with each tool). We chose to limit the mapping time to 30 minutes to reduce stress and time commitment for the participants. However, observing users for a longer period of time may lead to further insights when working with more difficult correspondences. Finally, although our participants were expert computer users and experienced with the domain of a university's structure, they were not experienced mapping users. The problems they experienced working through the tasks may be different than an experienced user.

5.5 Summary

In this chapter we presented the design and results of our first exploratory study on ontology mapping. This first study allowed us to observe users performing a mapping between two unknown ontologies using two unfamiliar tools in a controlled setting. The results gave us insight into what problems the users experienced using these tools, how the tools supported their mapping processes, and how they made decisions with the tools. In Chapter 8, we use results from this study to help develop a cognitive support framework for ontology mapping. In the next chapter, we continue this exploration into user-related concerns and issues by presenting results from an online survey with members of the ontology mapping community. This survey helps to address one of the limitations of the observational study, that is, the concern that novice mapping users may experience different issues than experienced users.

Chapter 6

Survey study

To design more effective tools and algorithms, we claim that a deeper understanding of the interplay between tool, user, and the process is needed. For example, who are the users that are going to use the mapping tools? Why do they need to perform mappings and for which domains? Do they use the currently available tools and if so, how do they use them? And, do these tools meet their needs? To answer these questions, we designed a survey and gathered feedback from the ontology mapping community.

In this chapter, we present the results from this survey. The information gained from this survey should be valuable to both tool and algorithm designers. The survey was also useful in discovering whether the problems our "novice" users experienced in the observational study were also experienced by more "expert" users. The survey was originally presented in [FNS07].

6.1 Survey design

We designed the questions, organization, and presentation of the survey with the help of several researchers with human-computer interaction and ontology-mapping experience. The survey consisted of **multiple-choice** and **open-ended** questions (Table 6.1). Ten of the questions were completely open-ended, three were multiple-choice, and five were multiple choice answers that also allowed an open-ended response.

The questions can be classified into three categories. Questions 1 through 5 were **user**

Table 6.1. *Survey questions*

No.	Question	Response
	User context questions	
1	What are the domains of ontologies that you work with?	open-ended
2	What is the average size of the ontologies you work with?	multiple-choice
3	What type of ontologies do you work with?	combined
4	How often do you create/edit ontology mappings?	open-ended
5	What are these mappings used for?	combined
	Tool questions	
6	What mapping tools have you used?	combined
7	Of the tools you've used, what do you find most useful and why?	open-ended
8	Of the tools you've used, what do you find to be deficient about these tools?	open-ended
17	If you were to design your perfect mapping tool, what features will it have?	open-ended
	Process questions	
9	Do you find it difficult to create an ontology mapping?	multiple-choice
10	If you answered "Yes" to question 9, please explain why.	open-ended
11	What process do you use when performing mappings?	combined
12	How do you remember which mappings you have created/verified versus mappings that are left to create/verify?	combined
13	When do you consider that the mapping is complete?	open-ended
14	Do you experience problems while performing ontology mappings? If so, please explain.	open-ended
15	How many people participate in the creation process of the mappings you perform?	multiple-choice
16	If the number of participants is greater than one, please briefly explain how the work is coordinated.	open-ended

context questions, questions 6 through 8 and 17 were **tool questions**, and questions 9 through 16 were **process questions**. The user context questions were designed to gather data relating to the use cases for ontology mapping. The tool-related questions were designed to explore our questions regarding which tools people are using and whether the users find the tools useful. The process-related questions investigated the difficulties with performing mappings, whether users are working in team environments, and what processes people are following for coordinating their mapping efforts. Table 6.1 lists the specific questions asked in the survey.

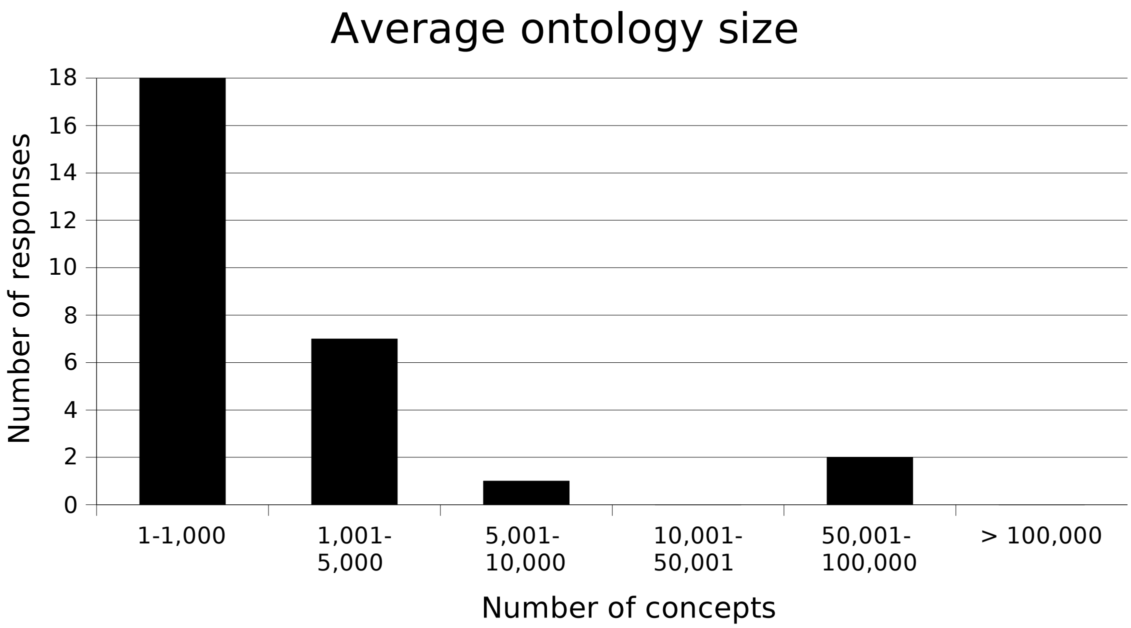

Figure 6.1. *Size of ontologies being used.*

6.1.1 Participants

The survey was available online for a two week period to both industry workers and academics. Respondents were recruited via semantic web related mailing lists and news groups (see Appendix D and E). Before starting the survey, informed constent was obtained (see Appendix F). Twenty-eight people responded to the survey.

6.2 Results

6.2.1 User context questions

The first question asked participants to describe domains of ontologies they work with. We had participants from a variety of domains, the most popular being biomedical, media, information-system design, business, and travel. Several respondents worked with different domains of ontologies, either for research purposes or based on their current project.

Figures 6.1 and 6.2 show the distribution of results for the questions on the size of the

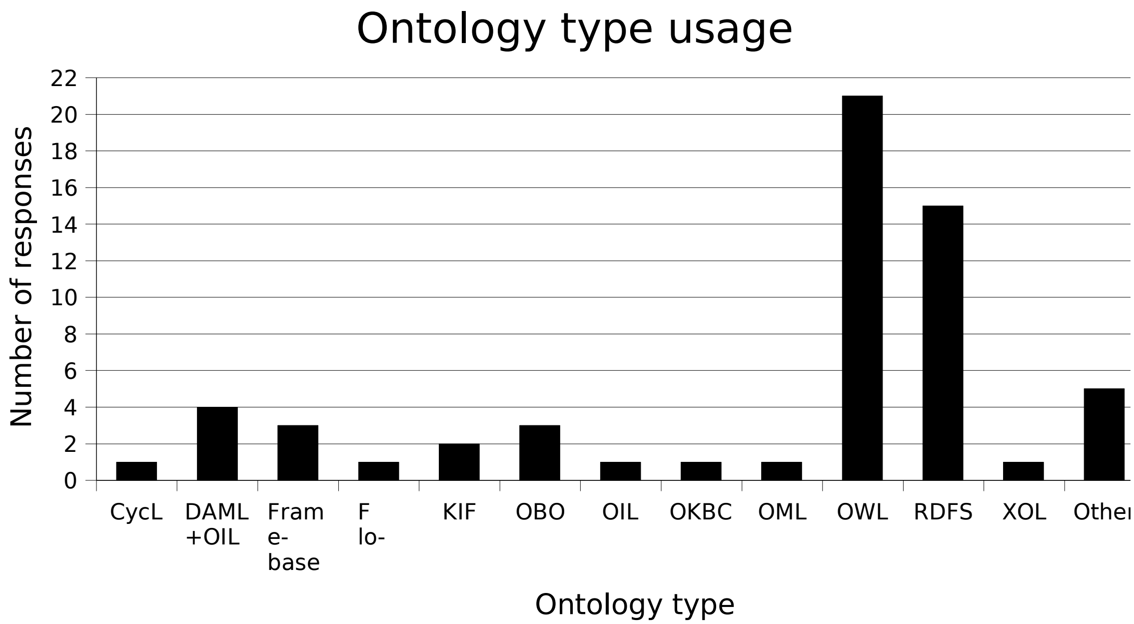

Figure 6.2. *Ontology language usage.*

ontologies and the ontology languages used. Most participants worked with ontologies with less than 1000 concepts and the primary ontology languages are OWL and RDFS.

In the next user context question, we looked at how often participants create and edit ontology mappings. The responses varied, the most popular being that mappings were created either per-application or as often as their work dictated that a mapping was required. Two of the respondents indicated they performed this operation often, but for research and tool testing purposes.

The final user context question asked why the participants created mappings. Participants were allowed to select from a list of responses (as many that applied) and also provide a free-form response in an “Other” field (see Figure 6.3). The use cases listed were “Instance data sharing”, “Ontology merging”, “Query translation”, “Web service integration”, and “Other”. Two of the “Other” responses stated that they create mappings purely for research purposes, while one was for automatic SQL generation.

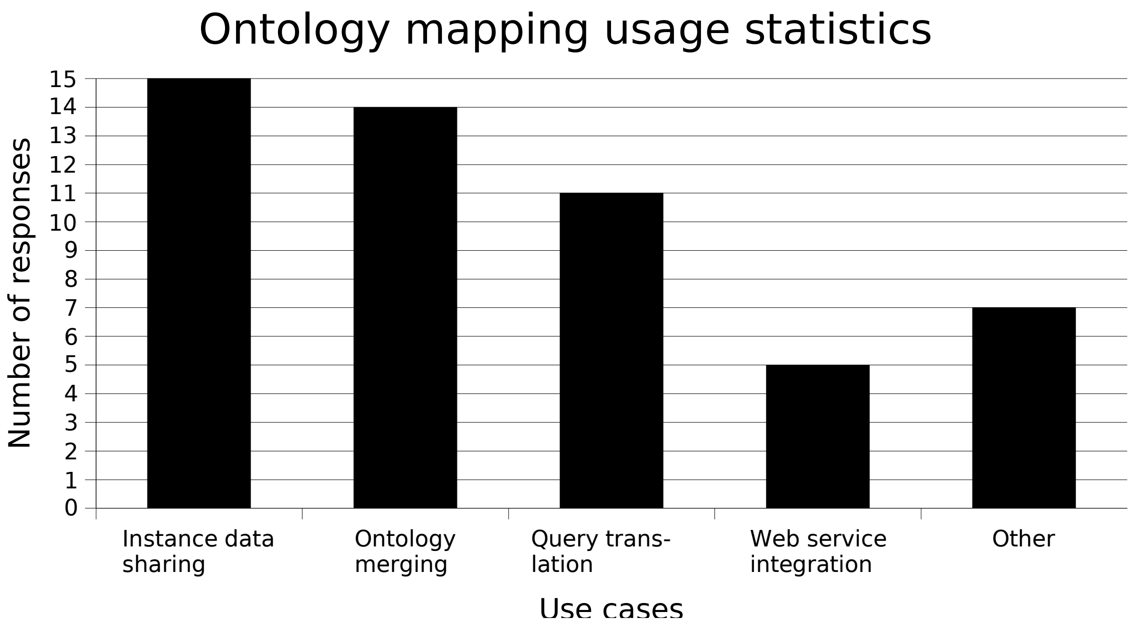

Figure 6.3. *Ontology mapping use cases.*

6.2.2 Tool questions

The first tool-related question asked which tools users had used. Respondents could choose from seven tools: Chimaera, COMA++, FOAM, MoA Shell, OLA, PROMPT, and QOM. They could also list any other tools in the "Other" field. Each of the listed tools was used by at most one to three participants with the exception of MoA Shell, which none of the respondents used. As shown in Figure 6.4, no tool was particularly dominant. The bulk of the feedback came in the "Other" category, which had 17 participant responses.

Other tools included Protégé, Internet Business Logic, AUTOMS, Crosi, WSMT with Ontostudio, X-SOM, OMAP, Falcon-AO, HMatch, and Snoggle. Each of these tools was used by only one participant, except X-SOM, which had been used by two. Two participants indicated that they use a custom built solution, while one indicated that they use a completely manual process.

We asked which tools and features participants found most useful and what deficiencies they found with the tools. Both Crosi and COMA++ were found to be useful because they integrate a large variety of similarity algorithms and are available online. One user indi-

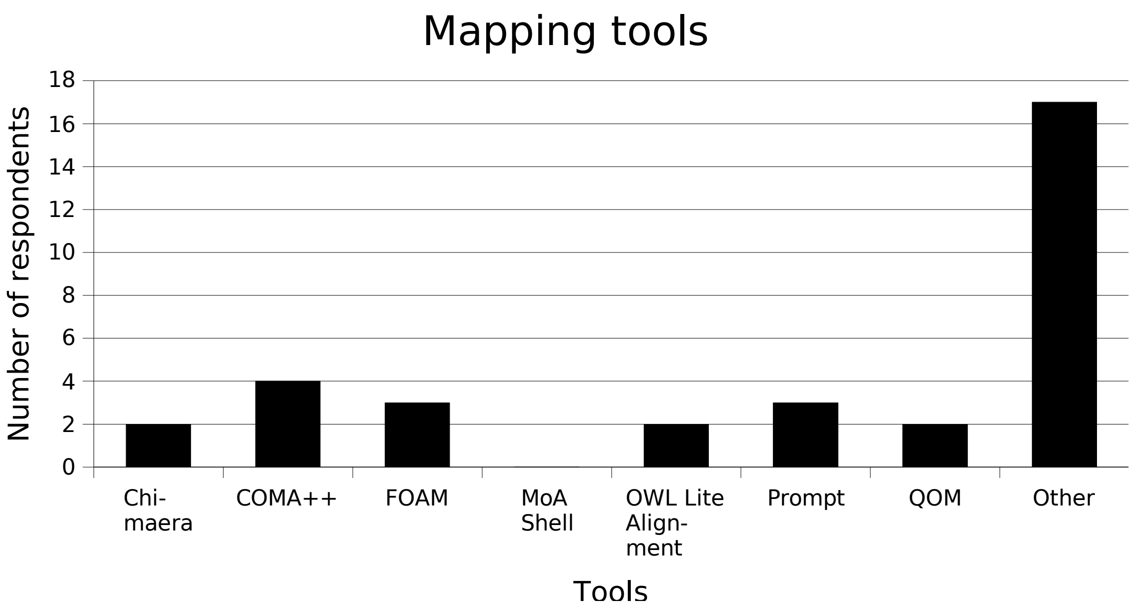

Figure 6.4. *Ontology mapping tools.*

cated that they like tools to provide simple suggestions and automatic help, while another user had a contrasting view, stating that they like statistically-based tools because others require too much designer opinion. Protégé was highlighted as being good for manual creation of a mapping as it makes it easy to create ontologies. Several participants pointed out that many tools are too general and are built without domain-specific mechanisms. One of the custom built solutions was indicated to be similar to PROMPT, but was designed to take advantage of domain knowledge, specifically term normalization algorithms and synonyms for their domain of interest. The requirement for the tools to incorporate domain-specific analysis and features was a common theme in response to several questions in the survey. Another common theme was the lack of visual displays or easy to use tools. Specifically, one participant indicated that PROMPT's interface was too complicated to give to a domain expert to do the mapping. One user criticized specific tools for their lack of documentation, for being buggy, and for not working as described. Other interesting observations were the lack of precision and recall for mappings in real world ontologies and that the tools do not allow for expressive enough mappings (e.g., some tools only support 1-1 mappings).

In the final tool-related question, we asked the respondents to describe which features the perfect mapping tool would have. In the presentation of the survey, this question came at the end, but here we categorize it as a tool question. There were several interesting themes that came up in the responses.

The first theme was that six of the 19 responses related to the desire for *better and easier to use tools*. Specifically, participants stated that they needed better interfaces, graphical cognitive support, improved user interactivity, and facilities for explaining manual correspondences. Users highlighted a large number of *desired features for the algorithms for generating candidate correspondences*: powerful and pluggable matching algorithms, recognition of re-occurring patterns of representing information, identification of not only simple correspondences but also of complex ones, and extending beyond mere word-pair associations and semantics. Four of the responses stated the requirement for perfect precision of recall for the mapping algorithms. Three participants also indicated that they want better facilities for *testing mappings and support for more expressive mappings*. The final interesting theme was *collaborative support*. Most of the respondents (76%) indicated that they work on their mappings in teams (see Section 6.2.3). Most available tools do not support this type of team development scenario.

6.2.3 Process questions

The first process-related question asked whether the participants found the creation of an ontology mapping difficult. 30.8% replied "No" to this question, while 69.2% said "Yes". The follow-up question to those participants that answered "Yes", asked participants to explain why they found the process difficult. Ten of the 21 responses discussed semantic issues, such as the process being too subjective or ambiguous. One participant pointed out that the *"semantics of the underlying ontologies are not usually well defined. Without a solid understanding of the semantics, it is almost impossible to perform the mapping correctly."* Respondents also discussed a lack of domain expertise for performing mappings, and that *"[y]ou have to get into the brains of the original developers of the ontologies being* *mapped."* Participants also stated that tools are not flexible enough for application-specific needs, resulting in the manual creation of mappings, which is neither scalable nor efficient. One participant indicated that the OWL primitives for expressing mappings are poor and that users are faced with making difficult decisions when two related concepts *"almost but not exactly match."* Three participants also indicated that problems with creating mappings resulted from poorly designed and documented tools.

We next asked participants what process they followed when performing mappings. Available responses were "Tackle the easy matches first" (37.0%), "Focus on a familiar area of the ontology" (51.9%), "Rely on the tool's ordering of mappings" (14.8%), and "Other" (22.2%). Responses for the "Other" category included performing an automated matching up front and then a debugging step, while two of the responses indicated that they first applied lexical, then structural, and finally semantic methods.

In the next question, we asked how the participants remembered correspondences they had created or verified. Most respondents chose from the provided answers, "The tool supports this" (37.0%) and "Paper and pencil" (55.6%), while 22.2% filled out the "Other" option. In the "Other" responses, one user indicated that the tool they use supports the process of remembering which correspondences have been carried out, which works well when mapping is done in a single pass, but extra help is needed for multiple passes. Another respondent indicated that they use their own codes to report the correspondences they create, which is similar to tracking the information by paper and pencil. Finally, one respondent indicated that they did not follow any good process. It is interesting that the majority resort to tracking this information manually by paper and pencil. Similar types of changes exist in software development and most integrated development environments and source control systems handle the tracking of this data. For example, the development environment Eclipse supports local history tracking. Every time a file is saved, the difference between subsequent versions is stored and can be used later for restoring the file to a given state or for comparing different versions.

We then asked when the participants considered the mapping to be complete. Ten of the 25 responses indicated that they used some form of testing (automated or manual) to verify that the mapping was completed to their satisfaction. For many respondents, this testing meant determining whether the mapping supports whatever application they were working on. Five responded that they knew the mapping to be complete when all concepts had been mapped. However, this implies either a perfect mapping, or that they knew when all reasonable concepts had been mapped. Interestingly, three participants responded that they never knew when the mapping was complete. Only one respondent indicated that they relied on tool support for determining whether the mapping was complete, although one participant stated that they must trust the system when mapping large ontologies because verification by hand is too slow.

We also asked participants about the types of problems they experienced while performing mapping operations. Similar issues outlined in previous questions came up again. Specifically, one respondent stated that *"most ontology tools are difficult for business users to understand."* Testing the mapping was also a popular theme along with issues related to the problem that people model conceptualizations differently.

The final two questions dealt with whether participants worked in teams and what sort of process they followed for coordinating their efforts. 53.9% indicated that only 1 or 2 people were involved in the mapping process, 42.3% worked in groups of 3 to 5 people, and finally, 3.9% worked with 6 to 10 people. Based on results from the team process question, we were able to determine that of the 53.9% working in teams of 1 to 2 people, 53.8% of these actually work in a team of 2, which means that only 24.0% work completely on their own.

The team-coordination processes ranged from weekly meetings to collaborating through wikis to coordinating through CVS. Eight-teen of the 20 respondents relied on non-software solutions for managing the team or a combination of *ad hoc* communication strategies like CVS, wikis and e-mail along with meetings. Smaller teams typically had one team lead and one implementer/mapper, and coordinated with face to face meetings. Participants also indicated that they worked as a group or partitioned the ontologies and then performed a group check to validate the mapping. Some teams used domain expertise for reviewing the composed mapping or during the mapping process for input. One respondent indicated that they use a “brainstorming” team process for coordinating the mapping effort.

6.3 Findings

We found it surprising how many tools had been tried by our respondents. There has clearly been a large effort from the research community to develop so many tools, yet there does not appear to be a dominant tool that is a benchmark for mapping tool design. This may be due in part to the variety of user needs. Some respondents highlighted that they had domain-specific needs or that existing tools do not support sophisticated enough mappings.

Most of the problems, deficiencies, and issues with ontology mapping uncovered by the survey can be classified into two categories: 1) fundamental issues with language and semantics, and 2) tool-specific issues. Fundamental issues, such as different model conceptualizations and language ambiguity, are difficult, if not impossible problems to solve. It is interesting that some of the responses to the “perfect mapping tool” question were that the tool would have 100% precision and recall or full natural language understanding. While a perfect, fully-automated solution would be ideal, it is probably not realistic for any but the most simple, straight-forward correspondences. As the survey also highlighted, many of the problems that people face in ontology mapping are difficult for even a team of human experts to resolve.

It is also interesting that these users felt that there could be an automatic algorithm that they would trust completely. Issues of trust also came up in our observational case study. Although our users stated that they liked PROMPT’s explanation facility, they were also often confused when it made a suggestion that was obviously wrong. Incorrectly generated candidate correspondences would sometimes lead to users ignoring the suggestions and switching to a completely manual process.

Tool specific issues such as better user interfaces, graphical support, better testing facilities (data translation based on a mapping), interactivity, algorithm explanation capabilities, and so forth, are all problems that ontology-mapping tool developers can help with. As discussed in the results, one respondent indicated that PROMPT was too complex to give to their end-users. This sentiment was also echoed by the non-computer science participant in the previous study. Mapping is a complex process, it is important that we do not further burden our users with learning a difficult to use software suite. Instead we must support their tasks with good software.

The responses to the process-related questions brought up several interesting issues that tool designers and researchers may also need to address. First, it is noteworthy that many of the participants relied on paper and pencil to remember the correspondences they perform. One individual even noted that some tools work for a first pass, but then they "forget" the steps previously performed during the second pass. Tool support should be able to address this issue. Second, it appears that most users work in small teams but tools currently lack support for team communication and collaboration, as well as for partitioning the mapping process into manageable chunks that can be tackled by individuals on the team. Many teams work together to validate and prepared the mapping. Again, tool support could help with team coordination. Metadata annotations (perhaps visualized via color-coding) could be used to record who composed the mapping and why they made certain decisions. In addition, the ontology mapping community could borrow ideas from the Web 2.0 social networking community [1]. E.g., tools could support voting on correspondences, commenting on and annotating correspondences, and associating instance data with a conceptualization. As discussed in Chapter 3, there has been some experimentation with community-driven ontology mapping [Zhd05, NGM08], but tool support is currently limited.

[1] http://del.icio.us, http://www.flickr.com

6.4 Limitations

There are of course limitations to this study, the first and foremost being the sampling size and population. Although we feel that 28 respondents gave us a wide variety of interesting and useful responses, with such a small sample it is possible that our responses are biased. Also, as we solicited participants from mailing lists, most of which were academically oriented, our sample may be biased towards researchers in the area rather than a balance between those working in research and industry. Finding and recruiting users from our target population was also an issue, because it is difficult to know how to best reach ontology mapping tool users.

As with any online survey, the wording of some of our questions may have potentially been confusing to some participants. For example, we asked "If you were to design your perfect mapping tool, what features will it have?". We stated that some respondents indicated full natural language understanding and 100% precision and recall. Perhaps if the question had been worded differently to solicit feedback on a "realistic ideal mapping tool" the responses would have been different.

6.5 Summary

We presented the design and results from a web-based survey. This second study provided insight into the global ontology mapping community. We learned more about what types of ontologies are being used, which applications people are creating mappings for, and the types of problems they experience. In the next chapter, we present a third study that focuses on team coordination and process with a specific team carrying out mappings in the biomedical domain.

Chapter 7

Interview study

The results from the survey were valuable in establishing a basic understanding of why people perform ontology mappings, the tools they use, problems they experience, and what processes, if any, they follow. However, the survey only gave us an overview of the mapping community. The results pointed out several important issues, but we could not ask our respondents specific follow-up questions. Also, our survey was primarily designed around which tools are being used and what are the advantages and disadvantages of these tools. We felt it would be useful to take a more in-depth look at user experience in this area, specifically looking at team dynamics.

In this chapter we discuss the design and results of our final exploratory study. In this study, we used interviews to investigate a specific team carrying out ontology mappings. As was seen in the survey results, most of our respondents worked on small teams. We wanted to better understand the process they go through to develop mappings, how they coordinate their efforts, what are their main problems, and how do they resolve them.

7.1 Study design

Our research questions reflect our desire to understand and investigate how users are working in teams, which process they are following, and which tools they are using. We attempted to address the following issues for a specific mapping team:

Q1: How does the team handle coordination?

Q2: What is the current team mapping process?

Q3: How are mapping correspondences verified?

Q4: What methodology is followed to carry out mappings?

Q5: Which parts of the mapping task are difficult?

Q6: What are the limits to their current tooling and how can the tools be improved?

We followed a qualitative research methodology. Our research consisted of background reading about the *SNOMED CT* terminology, conducting semi-structured interviews and participating in a *SNOMED CT* workshop hosted by the team. The workshop took place over two days and involved a number of students and researchers working on projects related to *SNOMED CT*. The researchers presented their work and discussions took place about how to better utilize the *SNOMED CT* terminology and how each of researchers could collaborate.

7.1.1 Participants

The team we studied works in Health and Information Sciences. The team has been working on creating mappings from medical domain terminologies to SNOMED CT (Systematized Nomenclature of Medicine – Clinical Terms) [SMD, SMD08]. SNOMED CT is a very large medical terminology (over 370,000 concepts) that covers most areas of clinical information like diseases, procedures, findings, micro-organisms, and pharmaceuticals. This team has been working to show the value of using terminologies like SNOMED CT by helping doctors and other medical personnel migrate their own domain-specific terminologies as well as previous standards like IDC-10 [1] to SNOMED CT via mappings. For the interviews, we selected three members (2 female, 1 male) of this team using convenience sampling [Cre03]. This is a sampling technique often used in exploratory research and in many practical situations. The members were in part selected because of their close proximity to our research labs, making the interviews easier to carry out. Recruitment took place through e-mail (see Appendix G) and informed consent was given (see Appendix H).

[1] `http://www.who.int/classifications/icd/en/`

7.1.2 Materials and Procedure

We used a semi-structured interview process, where we based our interviews on 18 open-ended questions, categorized into four different groups: introductory, team coordination, process related, and tool related questions (see Table 7.1). The interviews took between 30 to 60 minutes to complete, were audio recorded, and later transcribed. We chose this process as some of the members of the team are not co-located; thus interviews could take place over the telephone. In the course of each individual interview, some of these questions were omitted due to not being relevant or being answered during a previous question. Also, generally, follow-up questions were asked based on individual responses for clarification purposes or to explore something said by the participant.

7.1.3 Analysis

To analyze the data, we began by performing a preliminary exploratory analysis [Cre03] where we read over all the interviews and wrote summaries of the responses. These summaries were shared with the interviewees for verification. We used the experience from the workshop and background reading to help with the analysis of the interview data. Also, participation in the workshop helped to give us a better understanding of the domain and the problems the team is trying to resolve. Using the interview data, we conducted a coding process that involved segmenting and labelling the text from the interviews (see Appendix I). By consolidating our independent coding analysis and grouping codes we were able to establish themes that addressed our research questions.

7.2 Results

In this section we briefly describe the results from each of the interviews. We refer to each of our interviewees with the following fictitious names: Sarah, Rob, and Jessica.

Table 7.1. *Interview questions*

Introductory questions	
1.	Can you briefly discuss your educational and industry background?
2.	How much experience do you have working on mapping projects in these domains?
3.	Can you describe some of this experience? Goals, datasets used, etc.
Team coordination questions	
4.	Please describe the kinds of data mapping tasks you do in your job. Follow up: What role have you filled within these mapping projects?
5.	How many people have been involved in these mapping projects?
6.	What length of time did these projects take to complete?
7.	Can you describe the process that these projects have followed to build up a mapping? How do you coordinate the effort?
Process related questions	
8.	How do you decide what to map?
9.	How do you know when a given term does not have a potential mapping correspondence?
10.	When there are granularity differences, and you are mapping to a broader term, when do you stop looking for matches?
11.	How do you keep track of terms that have already been mapped?
12.	How do you keep track of where you have left off in between mapping sessions?
13.	When do you consider the mapping to be complete?
14.	What problems do you experience in a mapping project?
15.	How are the mappings saved and later used?
Tool related questions	
16.	What tools do you use to help you with mapping?
17.	What is most effective about these tools? Follow up: could you do the mapping task without these tools?
18.	If you were to design a better mapping process/tool, what would it include?

7.2.1 The Sarah interview

Sarah has worked in the health and medical field for 32 years, originally training and working as a nurse, later as management in hospitals and long-term care facilities, and then policy and analysis work for the government. She started a graduate degree in Health and Information Sciences several years ago and began working with SNOMED CT. Her primary project has been to map a subset of ICD-10 terms to SNOMED CT. ICD-10 has around 17,000 terms, but for this project, a subset of 5,000 terms were selected. This subset represented 95% of the terms used in a diagnosis and discharge dataset that was analyzed.

This project was structured fairly loosely, based on interest and individual time lines.

Each member of the team had individual priorities that needed to be met. According to her, there has been very little work with SNOMED CT to support the electronic health record and map various existing terminologies. Part of Sarah's role was to help figure out the process for actually developing these mappings and what should be achieved through the mappings. These questions have not been addressed in the literature and there is no existing guide on how to map terms to SNOMED CT.

As a result of over a year of work, off and on, the team ended up with a subset of terms that were mapped one-to-one. These mappings were discovered by a tool that Rob developed. The tool primarily relies on lexical similarities for determining matches and categorizes outputs as exact matches, partial matches, and no match found. They are now struggling with how to actually assess the quality of this mapping. One of the primary issues is that some terms come up with many possible matches (20 or more) and these are difficult to evaluate. Sarah and other domain experts in the group have to manually evaluate these matches, but their tooling has not been developed to support this process. Sarah states that in order to evaluate a correspondence, you need to know where a term fits into the concept structure, how the pieces fit together, and that it is better to know more information rather than less. This, along with further development and refinement of their algorithms is among their biggest challenges for the future.

7.2.2 The Rob interview

Rob has a background in computer and information systems rather than health care. His primary role in the project was to develop the software used to automatically discover mappings. The software is web-based and works by allowing users to upload bulk lists of terms. These terms are compared to their SNOMED CT indices for matches and these results are downloadable in an Excel spreadsheet. The only structural information they use from SNOMED CT is the historical relationships, which relate terms to older variations. Rob indicated that using the subclass hierarchy would lead to too many results to analyze.

Rob has also written data extraction and cleaning tools. The first step in any of their mapping projects is to first extract the data from its current storage repository. This data is processed through their batch online tool to get the mapping results. By inspecting the results of terms with no matches found, they sometimes determine that a cleaning process is necessary to possibly remove noise such as underscores, abbreviations, spelling differences, etc.

The team as of yet does not have a formal validation process. However, they do attempt to validate each other's work by manually inspecting mapping correspondences that have been generated by an individual. Generally, the expectation is that all terms should be mappable to SNOMED CT as this is the most comprehensive medical terminology in existence. Validation is always performed manually by a domain expert. Depending on the project, the expert is either an existing member of the group or possibly the creator of the terms.

Rob stressed that in the future they need to improve their tools for constructing more complex matches, such as one-to-many correspondences. These matches cannot easily be discovered automatically, but they need tools to support the validation and creation of these mappings. Also, reporting is an important step in their process. They need reports for discussing specific mappings as well as summarizing results. Finally, Rob and his colleagues are working to standardize their process for generating mappings. Currently, much of the process is manual, application specific, and exploratory.

7.2.3 The Jessica interview

Jessica originally trained and worked as a medical doctor and has been working in the health and medical field for more than 10 years. Currently, she is working in the computer science and health and information science field. Independently to Rob, she has also developed a mapping tool. She also applies her domain expertise to help discover and validate correspondences. Her primary tool for this process is searching SNOMED CT with the tool CliniClue [2]. CliniClue allows her to find correspondences one term at a time. Any

[2] http://www.cliniclue.com/

correspondences she discovers are then recorded in a database.

Her mapping discovery is largely based on her experience and expertise. She uses the structure of SNOMED CT to explore related terms and to help determine the best match. Similar to Rob, she has performed data cleaning. Due to her domain expertise, she can often search for synonyms of terms or use search engines like Google to discover alternative expressions. The cleaning process may also involve segmenting or splitting the original expressions if they are too long for mapping. In this process, context is very important. Her work is later validated during team meetings or by her supervisor. It is an iterative process where discrepancies are discussed and compared until an agreement is reached.

One of her primary challenges has been in trying to encode local terminologies with SNOMED CT. Not all the terms are clinical expressions. SNOMED CT is very specific, but often the local terms are very general. Similar to Sarah, she struggles to validate terms that have a large number of potential correspondences. To help alleviate these issues, she feels that they need to develop standards for mapping local terminologies and establish a standard format for storage of the results. The terms need more detailed descriptions and documentation about their intended usage. Jessica anticipates that their future tools need to be able to support plugins that can be integrated for specific functionality corresponding to the terminologies being mapped. Also, the tools need to natively support searching and navigation of SNOMED CT.

7.3 Findings

A number of themes emerged from our analysis. Below, we discuss these themes and link them back to our original research objectives. The themes are categorized into three groups: 1) Team coordination/process; 2) Mapping process; and 3) Tool limitations and demands. Note that although these findings are categorized into three groups, there is overlap and interplay between the categories.

7.3.1 Team coordination/process

Our first three research questions ask about team coordination, process, and mapping verification. The following themes address these questions.

7.3.1.1 Diverse backgrounds necessary

The team consisted of members with a variety of diverse backgrounds and Sarah stated that she felt this was important for any team working in this area. *"A team of maybe five people with cross program skills, it can't be all computer scientists type people . . ."*. We see from the interviews that various members of the team relied on each other's unique expertise, such as domain knowledge and programming skills. Like complex research projects, large mapping projects rely on a variety of skill sets coming together to make the project work.

7.3.1.2 Developing a methodology

This team is working at the forefront of research centred around SNOMED CT. As a result, they have been responsible for developing a methodology, process, and tooling to carry out mappings. Developing this methodology was considered one of the central challenges from all three participants and all felt that further work towards standardizing this was critical to the process.

The team worked together to develop their own tooling to help them discover mappings. Creating a mapping is an iterative process, involving a number of steps such as data extraction, cleaning, re-factoring, team validation, and communication with domain experts.

7.3.1.3 Cooperative validation

The team worked together to help validate each other's work. This consisted of weekly meetings where they would discuss particular issues and resolve differences. They worked with domain experts, either within or outside their team, such as the creator of a terminology. This cooperation was their primary process for assessing and evaluating the quality of their results.

7.3.2 Mapping process

In this section we discuss themes that relate to the mapping and validation process, which helps to answer research questions four and five.

7.3.2.1 Simplify first

The interviewees stressed the importance of simplifying the task of mapping first. For example, in the ICD-10 project, the first step was to reduce the number of terms down to the most critical subset of 5,000 terms. Following this, they used an iterative process of running it through their tool, inspecting the outcome, cleaning the data, and re-running the algorithms. The team generally accepted exact matches as correct in an effort to reduce the number of correspondences that needed to be explicitly checked. These steps are necessary for helping to deal with the shear size of these datasets. As Sarah stated, *"[T]he scale is always a problem, because the potential scale is so large."*

7.3.2.2 Series of iterations

Each of the interviewees discussed the process of mapping in terms of iterations. Jessica stated, *"I think what the tool needs to become is something where you can apply various steps and at each place examine the output"*. The tool needs to allow a user to make certain assumptions, such as exact name matches are correct, in order to help reduce the complexity, and allow the human to focus on more complex matches. Moreover, a clearly defined series of steps helps establish completeness or correctness. If a term fails to have a match after an established set of steps have been taken, then that negative result potentially yields insights into deficiencies with the methodology.

7.3.2.3 Difficult mappings

The interviewees considered the validation and creation of one-to-many mapping correspondences to be amongst the most difficult to determine. *"What do you do with the terms that came up with 20, 30, 40, 50, 100, 200 different possible maps? And how do you even think about that?"* SNOMED CT is such a large terminology that it is difficult to know when to cut off a particular correspondence, especially if the local terms are very general and widely applicable expressions. Currently, they have no tool support or process for this. It is one of the central challenges they are facing in the near future.

7.3.3 Tool limitations and demands

The final set of themes relate to tool specific issues and help address research question six, *"What are the limits to their current tooling and how can the tools be improved?"*

7.3.3.1 Existing tools fail

The team chose to develop their own tool due to a lack of openly available free tools. They also use the CliniClue browser for SNOMED CT, however, this tool was developed for searching and browsing SNOMED CT, not mapping. As a result, it is not suitable for computing correspondences with a set of terms. Also, since SNOMED CT is such a large ontology, many existing research tools in the ontology mapping community are not suitable.

These results are consistent with results from our online survey. Many respondents of the survey indicated that they chose to develop their own tools as well. This was partly due to the general nature of many of the research tools. Often, people need domain specific tools that are tailored to their specific needs, as our interviewees indicated.

7.3.3.2 More detail required

Their existing tool needs to produce more detailed output to aid in validation. Currently, results are reported in an Excel spreadsheet. There is no direct way to inspect the SNOMED CT term's structure, which all three interviewees indicated as very important.

7.3.3.3 No methodology is flawless

The interviewees indicated that no methodology/algorithm is flawless and that a user always has to be part of the process. New development for data extraction, cleaning, and validation often takes place on a per-project basis. Automatically discovering and generating one-to-many relationships is too complicated, and currently the responsibility of working through the large number of possible matches is left to domain experts. Also, relying solely on the category labels to determine matches will potentially introduce bias as was discussed in the experiments on human inference (see Chapter 3).

7.3.3.4 Reporting

Reporting is essential to their mapping process. They need to produce detailed summary reports about their mapping results to share with the other group members and share with their clients. These reports can act as a paper trail for comparing new results generated after a cleaning or algorithm tweaking phase.

7.4 Limitations

The main limitation of this study was that we limited the investigation to a single mapping team. As a consequence, the results may not be applicable to all teams. However, we do see consistencies between the results of this study and our larger and broader survey study. Another potential limitation is the maturity of the team we studied. In comparison to the experience reported by Reed *et al.* [RL02], which was developed over 15 years of mappings, the team we studied was relatively new to mapping. There may also be a bias in the interpretation of the results and theme analysis. Results from our previous two experiments may have influenced our choice of codes and themes for this experiment. We attempted to mitigate this by having one researcher that was unfamiliar with the previous two experiments help code the interviews. Despite these limitations, we feel the interviews provide rich insights into some of the challenges of mapping large ontologies and how in particular, one team has been working to resolve these challenges.

7.5 Summary

In this chapter we presented the design and results from an interview study that focused on team-related issues during ontology mapping. We presented a number of findings based on this experiment, which help contribute to our understanding of ontology mapping issues as well as highlight potential opportunities for cognitive support related to team process.

This study also highlighted that ontology mapping is much more than the algorithms used to compute matches. This is but a small part of the entire process. Mapping consists of understanding the data you are working with, establishing a means of processing it, cleaning or standardizing it, iterating through a mapping procedure, and working in a team to validate and assess the quality. This is an iterative process and one where the human or humans are essential contributors. Tool developers can help by creating tools that support domain specific needs. We believe that developing new techniques that help reduce the *accidental complexity* [PB85], while still allowing people to deal with the *essential complexity* [PB85] of the problem along with ways to visualize and explore potential correspondences, would greatly speed up the analysis and validation procedure. The tool must support the mapping process, not be a hindrance to it.

In the next chapter we combine results from the three experiments and the results from Chapter 3 to present a set of information needs of ontology mapping users. Based on these needs, we propose a process model that describes the opportunities for cognitive support for ontology mapping systems.

Chapter 8

A cognitive support framework

In this chapter we synthesize results from the three exploratory studies and the work from Chapter 3 on human inference to address research objectives **O5** and **O6** (see Chapter 1 Section 1.2). We first present a set of information needs that were identified through our analysis. These needs are represented as questions that are important for the user to answer during ontology mapping. Following this, we present a process model that details the various opportunities for cognitive support in ontology mapping systems. This model is first presented without including the team aspect of ontology mapping and then is extended to include our team-based results. Using this model, we derive design requirements for incorporating elements of cognitive support into ontology mapping systems.

The combination of the information needs and models represent the cognitive support framework for semi-automatic ontology mapping. We use the term "framework" in this context to refer to a "theoretical framework", which helps to make logical sense of the relationships, variables, and factors that we have discovered to be important [Bor96, Bot89]. The framework provides a frame of reference for communicating our findings, as well as helps to guide future research.

Parts of this chapter are presented in [FS07b].

8.1 Information needs

We analyzed the data from each of the studies and identified a set of information needs that are important to ontology mapping users. These needs are shown below in the form of questions that a user may need to answer.

1. How do I know when the mapping procedure is complete?
2. How can I verify the quality of my mapping?
3. How can I identify similar areas of the ontologies?
4. How can I limit the scope of the mapping?
5. How can I understand why a particular correspondence is suggested automatically?
6. How can I make temporary decisions and reverse decisions about mapping correspondences?
7. When mapping an ontology, where should I start?
8. How do I flag or indicate a questionable correspondence?

These information needs represent different opportunities for cognitive support in ontology mapping systems. Tools must support users through the process of answering each of these questions. In the next section, the semi-automatic mapping process model we propose addresses these user information needs within the context of a human-machine symbiotic mapping system.

8.2 Cognitive support opportunities

We first introduce the individual process model for semi-automatic ontology mapping. This model describes the relationship between the user and the tool and the principles of the model represent opportunities for cognitive support.

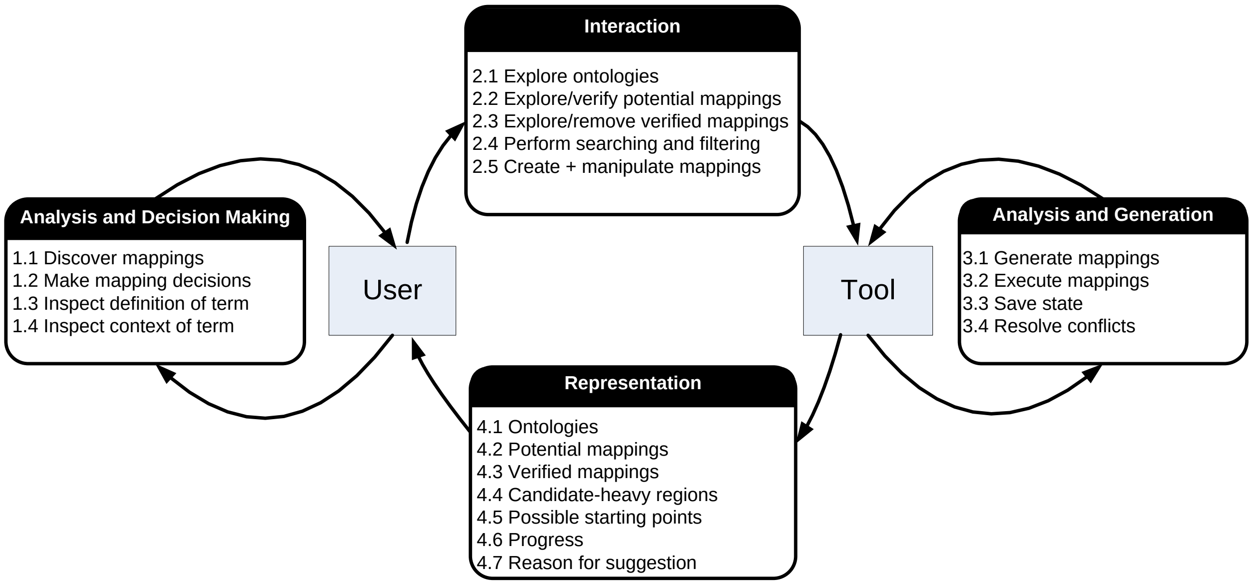

Figure 8.1. *A theoretical framework for cognitive support in ontology mapping.*

8.2.1 Individual process model

The model shown in Figure 8.1 has four conceptual dimensions: *User Analysis and Decision Making*, *Interaction*, *Analysis and Generation*, and *Representation*, which are based in part on work from Détienne [D0́2, pp.7], Victor [Vic05], and Walenstein [Wal02c]. Each dimension represents a concept in the human-guided ontology mapping process. Users internally perform analysis and decision making to understand and validate correspondences. Human inference plays an important role during this process. Externally, the users interact with the tool to acquire information or create correspondences. The tool internally performs analysis and generates correspondences and externally presents these to the user. Distributed cognition between user and artifact (tool) makes the task manageable.

In a mapping scenario, a user begins by choosing two ontologies to map by interacting with the tool. The tool provides representation of these ontologies and possible mapping correspondences. The user performs steps to verify the correspondences and then creates the mapping. This information is communicated back to the tool through user interaction, and the tool can use this information to update its representation. This iterative process continues until the user decides the mapping is complete.

Interaction is how the user communicates with the tool. Mapping users must be able to explore the ontologies and correspondences. This activity reinforces learning of the terms in the ontologies and involved in the mapping. Interaction also aids discovery of new correspondences. *Representation* is a closely related concept to interaction. While interaction is concerned with the exploration of data, representation is the presentation of data sources and the mapping. Representation of the ontologies and mapping supports redistribution, as this information does not need to be stored and tracked by the user. Progress indication and identification of starting points provides an ends-means reification for the user. The combination of these two dimensions may afford perceptual substitution, where an interactive structural representation of the mapping and ontologies allows the user to rely on their fast visual processing for exploration.

These are also important concepts related to user understanding and planning procedure. How the data is presented is critical to the problem solving aspect of mapping. Zhang *et al.* [ZN94] conducted a problem solving user study where participants were given multiple versions of the same problem. Each version used a different representation. It was shown that how the problem was represented was directly related to the difficulty of the task. Also, results from the ontology mapping observational case study demonstrated that mapping representations relate to the planning and attack strategy used by the tool user. We also saw a similar phenomenon in the experiments on human inference. Pictorial class representation was shown to have less influence over inference than textual labels.

User Analysis and Decision Making facilitates the justification of a mapping. There must be means for the user to verify that a correspondence is correct. Otherwise poor decisions could be made due to the user lacking a full understanding of the terms involved, properties of those terms, and relationships of those terms. Finally, the *Analysis and Generation* dimension is the generation and manipulation of mapping correspondences. This is a critical concept in the mapping procedure, as without it, no mapping can be created.

The model dimensions are described below along with descriptions of each model principle. We use the principles to derive software tool requirements, which we use to guide the development of our own mapping system. We also indicate the source of each principle, that is, which experiments the principle was derived from.

Analysis and Decision Making

(#1.1) Discover mappings:

Model Principle: Users discover mapping correspondences based on their domain knowledge or by exploring the ontologies. This information is often internalized until the user is convinced of the mapping.

Requirement: Support ontology exploration and manual creation of mapping correspondences. Provide tooling for the creation of temporary correspondences that the user can address at a later time.

Source: Observational case study and interview study.

(#1.2) Make mapping decisions:

Model Principle: Users internally make mapping decisions. The tool aids this by suggesting potential correspondences that the user validates.

Requirement: Provide a method for the user to accept/reject a suggested correspondence.

Source: Observational case study and interview study.

(#1.3) Inspect definition of term:

Model Principle: The definition of a term comes from the properties that describe the internal structure of the term. The internal structure helps explain the meaning of the term, which facilitates the user's understanding of the ontology.

Requirement: Provide access to full definitions of ontology terms.

Source: Observational case study and interview study.

(#1.4) Inspect context of term:

Model Principle: Context is how a term is used in an ontology. This is derived from the external structure (the `is_a` hierarchy) and the internal structure (definition of the term).

Context of terms in a mapping helps the user verify that the intended meaning of terms is the same.

Requirement: Show the context of a term when a user is inspecting a suggested correspondence.

Source: Observational case study, survey study, and interview study.

Interaction Dimension

(#2.1) Explore ontologies:

Model Principle: User-driven navigation of terms, properties, and relationships in the ontologies enforces understanding of the ontology and discovery of correspondences.

Requirement: Provide interactive access to source and target ontologies.

Source: Observational case study, survey study, and interview study.

(#2.2) Explore/verify potential mappings:

Model Principle: Exploring potential correspondences aids the user in the verification process.

Requirement: Support interactive navigation and allow the user to accept/reject potential correspondences.

Source: Observational case study, survey study, and interview study.

(#2.3) Explore/remove verified mappings:

Model Principle: Navigation of the verified correspondences allows the user to explore what they have completed and what is left to complete.

Requirement: Support interactive navigation and removal of verified correspondences.

Source: Observational case study, survey study, and interview study.

(#2.4) Perform search and filter:

Model Principle: Search and filter facilitates the reduction of information overload for mapping. It also facilitates planning as they allow the user to focus on smaller chunks of the mapping process.

Requirement: Provide support for searching and filtering the ontologies and mapping (e.g., filters to display terms in the ontologies with/without correspondences, or display only the correspondences with exact name matches.)

Source: Observational case study, survey study, and interview study.

(#2.5) Direct creation and manipulation of the mappings:

Model Principle: Many correspondences are missed by automated procedures, requiring the user to manually create them. Manipulation refers to adding metadata to a verified correspondence, such as a reason for the mapping.

Requirement: Support for adding details on verified correspondences and manually create correspondences.

Source: Observational case study, survey study, and interview study.

Analysis and Generation

(#3.1) Generate mappings:

Model Principle: Automatic generation of correspondences helps users identify simple mapping correspondences.

Requirement: Support the automatic discovery of some correspondences.

Source: Observational case study, survey study, and interview study.

(#3.2) Execute mappings:

Model Principle: Executing a mapping is the process of transforming instances from one ontology to another based on the mapping definition. This can be treated as a *debugging* step in creating a complete mapping: the user can verify if the instances created in the target from the source instances are the ones that (s)he expected.

Requirement: Allow the user to test a mapping by automatically transforming instances from the source to the target ontology.

Source: Survey study.

(#3.3) Save verification state:

Model Principle: Automatically saving the mapping state and returning to that state with each session relieves the user's working memory from determining where they were, what they were doing, and what their next step is, after an interruption.

Requirement: The verification process must support potential interruptions by automatically saving and returning users to a given state.

Source: Observational case study and survey study.

(#3.4) Conflict resolution and inconsistency detection:

Model Principle: Conflict resolution helps users determine inconsistencies in the created mapping. They can arise from a variety of situations, such as when two concepts are mapped, but some structural elements that are critical for their definition have not been mapped yet.

Requirement: Support identification and guidance for resolving conflicts.

Source: Survey study and interview study.

Representation Dimension

(#4.1) Source and target ontologies:

Model Principle: Representation of the ontologies facilitates understanding and discovery.

Requirement: Provide a visual representation of the source and target ontology.

Source: Observational case study, survey study, and interview study.

(#4.2) Potential mappings:

Model Principle: Representation of a potential correspondence aids the discovery and decision making process.

Requirement: Provide a representation of a potential correspondence describing why it was suggested, where the terms are in the ontologies, and their context.

Source: Observational case study, survey study, and interview study.

(#4.3) Verified mappings:

Model Principle: Representation of verified correspondences frees a user's working memory from remembering what they have already verified.

Requirement: Provide a representation of the verified correspondences that describe why the correspondence was accepted, where the terms are in the ontologies, and their context.

Source: Observational case study, survey study, and interview study.

(#4.4) Identify "candidate-heavy" regions:

Model Principle: Identification of candidate-heavy regions aids the planning procedure for performing mapping operations. It also facilitates understanding of results from the automated procedure.

Requirement: Identify visually candidate-heavy regions based on the automated mapping procedure.

Source: Survey study and interview study.

(#4.5) Identify possible starting points:

Model Principle: A starting point represents an area of the ontologies or potential correspondences where the user may wish to first concentrate their mapping effort.

Requirement: Indicate possible start points for the user, e.g., flag terms that have exact name matches, as these are generally the most straight-forward correspondences to perform.

Source: Observational case study and survey study.

(#4.6) Progress feedback:

Model Principle: Progress feedback facilitates planning, as it provides details about where the user is in the overall mapping process. This is also an indicator about the current verification state.

Requirement: Provide progress feedback on the overall mapping process.

Source: Observational case study and survey study.

(#4.7) Reason for suggesting a mapping:

Model Principle: Mappings auto-generated by the tool can support verification and understanding by "explaining" why the algorithm decided the two terms match. An explanation facility helps the user to decide on a correspondence and also builds trust between the algorithm and the user.

Requirement: Provide feedback explaining how the tool determined a potential correspondence.

Source: Observational case study, survey study, and interview study.

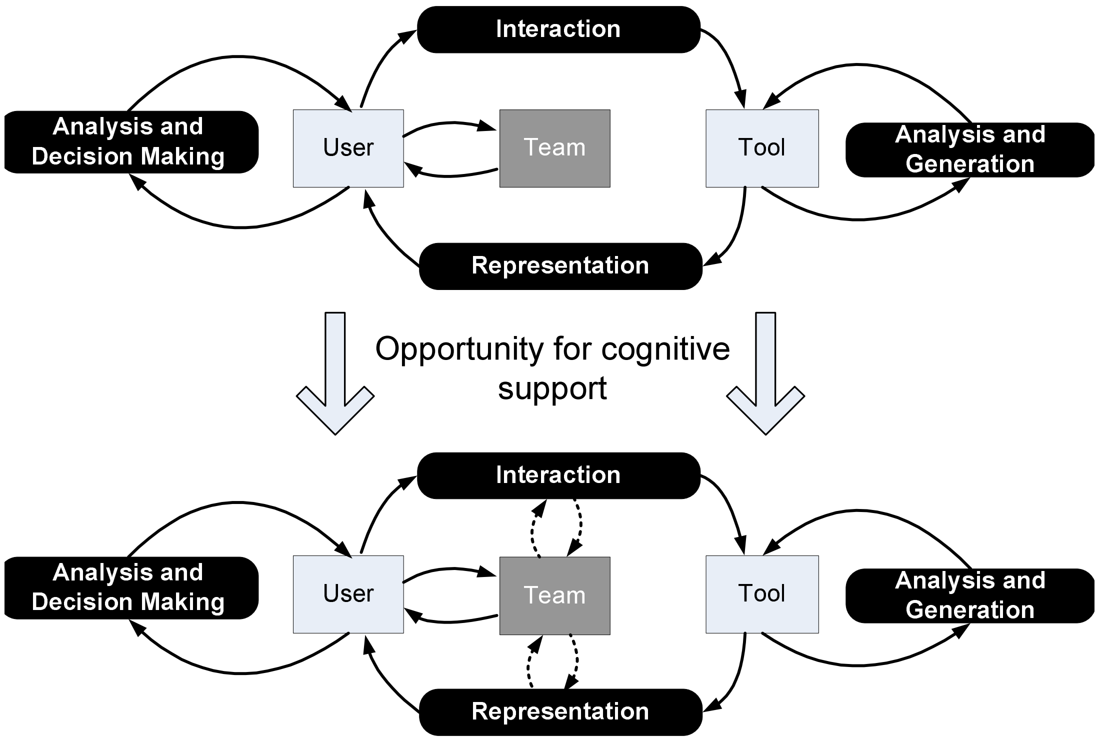

Figure 8.2. *Opportunities for cognitive support in team process.*

8.2.2 Team process model

In this section, we extend the "individual process model" to incorporate cognitive support elements for team process. Most ontology mapping tools do not support a collaborative team process, yet results from our survey and interviews indicate that many mapping projects consist of small to medium size teams.

In Figure 8.2, the top diagram, based on the "individual process model", encapsulates the process our interviewees followed in their own work. Here, we add the team dynamic to the diagram. Our interviewees and survey respondents indicated that team interaction primarily occurs through face to face communication or asynchronous communication such as e-mail and wikis. We feel there is an important opportunity for cognitive support by incorporating team interaction more closely into the process as shown in the bottom diagram (defined by the dotted arrows to and from Team). In this process, we do not completely remove user to user communication, but communication can also take place through interaction with the system and the representation the system provides.

Users need to be able to associate metadata with the correspondences they generate so that they can explain why they made certain decisions or highlight areas that need discussion or validation. This metadata can also act as a paper trail for their decision-making process. Users of the mapping can then understand why those decisions were made. Team-member based filtering should be introduced where a user can filter to show the correspondences created by a given member of the team and have awareness about what that member is currently working on.

The tool should support annotation of terms or concepts. Part of the mapping process is gaining an understanding of the source and target concepts. Large ontologies like SNOMED CT have external documentation describing concepts and their usage. However, users should be able to further explain the concept directly within the tool in order to communicate and document their discoveries. This helps to develop the team's shared understanding.

This level of team support is now available in software development. Sophisticated tools like IBM's Jazz [JZZ], have been developed with software collaboration and awareness as a central component. The tool suppports a process where developer activity is communicated with other members of the team and managers are able to know what people are working on. These ideas can also be incorporated into ontology development.

Ontology mapping can also begin to borrow ideas from social computing. Wikipedia[1], YouTube[2], and Flickr[3] depend on massive contributions and collaboration from their users. Similar value could be obtained for ontology mapping by providing tools where decisions can be shared, commented, and policed by communities of users.

[1] `http://www.wikipedia.com`
[2] `http://www.youtube.com`
[3] `http://www.flickr.com`

8.3 Summary

In this chapter, we introduced a cognitive support framework for ontology mapping that consists of user information needs and a model that describes the relationship between user and tool in the mapping process. This framework is based on a combination of research: literature and studies on human inference, theories of decision making, and three exploratory studies. The framework highlights the importance that both the user and tool play during the mapping procedure. We also see that the automated generation of mappings, which has received the most research attention in this area, is a small part of the entire mapping process. Moreover, based on our studies, the problems users experience go beyond the processing of the algorithms. Users have trouble remembering what they have looked at and executed, understanding output from the algorithm, remembering why they performed an operation, reversing their decisions, and gathering evidence to support their decisions. We believe addressing these problems is the key to improving the productivity of the users.

In the next chapter, we describe our ontology mapping tool COGZ. The tool is based on the principles and requirements outlined in this chapter.

Part III

Applying and evaluating the framework

Chapter 9

The CogZ tool

In this chapter, we use the requirements derived from the cognitive support framework to design a tool for ontology mapping, addressing research objective **O7**. We discuss the various iterations of the tool design, the various features, and the rationale for the existence of each feature. We also relate each feature to the framework requirements. We primarily concentrate on the features of the individual process model. Parts of this chapter were originally published in [FYS].

9.1 Towards cognitive support

Rather than building a tool from scratch, we decided to extend an existing mapping tool with a plugin for cognitive support. We recognized PROMPT as the best match for our cognitive support tool integration. This is because PROMPT already addresses some of the cognitive support requirements we defined and it is available as an open source tool. By working with the PROMPT developers, we created an extensive plugin architecture that allows researchers to easily plug-in their own algorithms, user interface components, and mapping file formats. Using this plugin architecture, researchers can extend many of PROMPT's user interface components. These extensions to PROMPT were first discussed at the "Ontology Matching Workshop 2006" [FNS06].

We decompose the mapping process into steps: an algorithm for comparison, the presentation of mapping correspondences, fine-tuning and saving of correspondences, and exe-

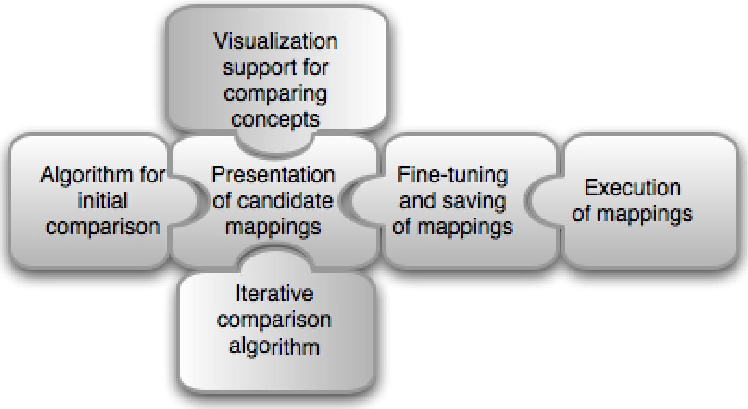

Figure 9.1. *Configurable steps in the* PROMPT *framework. Developers can replace any component in the figure with their own implementation. Each component has opportunities for cognitive support and tool support features can be introduced at each step to aid cognition.*

cution of a mapping (see Figure 9.1). These represent plugin extension points in PROMPT. These extensions allow researchers to move their ideas from prototypes to fully implemented mapping tools, without recreating the entire user interface. These extensions to PROMPT provide the ontology engineering community with a consistent interface for mapping and allows users to have access to a suite of tools and algorithms. In Figure 9.2, some of the various UI extension points that are now supported by PROMPT are shown.

There have been a number of plugins developed for PROMPT based on the supported extensions. As a "proof of concept", we developed an algorithm plugin for the FOAM [ES05] mapping library. This plugin acts as a "bridge" between PROMPT and FOAM. An algorithm plugin has been developed for using UMLS as well as one for matching class name synonyms. Both are included with the default installation of PROMPT. Also, as part of the default installation, there are two plugins available for storing a mapping: one that uses a simple mapping ontology and one based on the problem solving methods described by Crubézy [CM03]. Storage plugins give PROMPT flexibility to support any of the mapping relations discussed in Chapter 2, Section 2.3.

The plugin framework allowed us to design our tool as a user-interface (UI) plugin

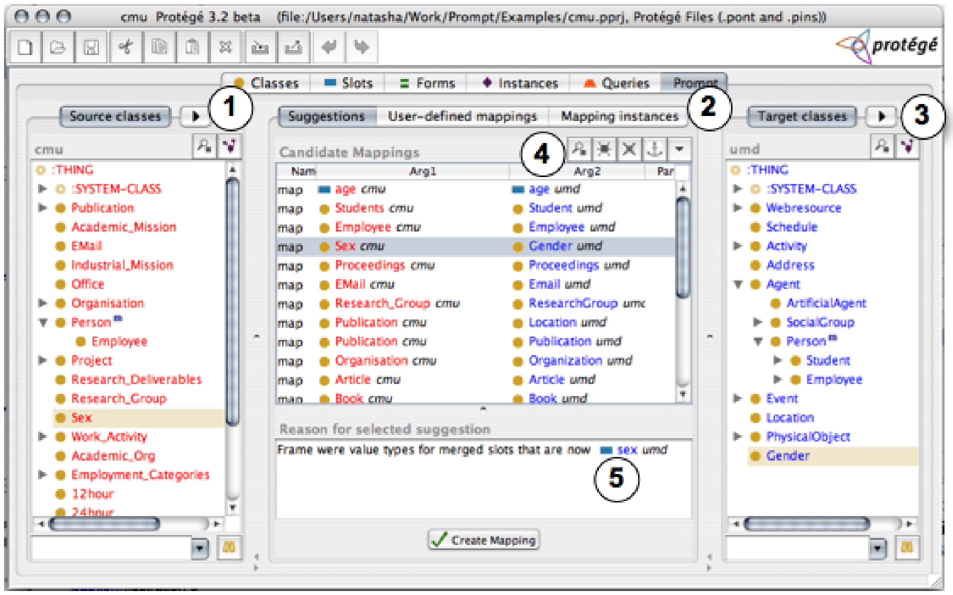

Figure 9.2. *The* PROMPT *user interface and the extension points in* PROMPT*'s mapping component. The left column shows the source ontology; the middle column displays the correspondences suggested by* PROMPT *and explanations of these suggestions. The right column displays the target ontology. There are tab extensions points for the source (1), mapping (2), and target (3) components. Area (4) shows the suggestion header button extension point. Algorithms can provide their own explanations for each candidate correspondence (5).*

focusing on cognitive support. We developed a PROMPT plugin called COGZ (Cognitive Support and Visualization for Human-Guided Mapping Systems). COGZ was first introduced in 2006 [FNS06], a second version was released in 2007 [FS07b], and finally the latest version was released in 2008 [FYS]. Because COGZ works as an extension to PROMPT, it can harness the features of PROMPT and enhance or support them with additional visual components. The plugin architecture also allows any algorithm plugin to indirectly benefit from the cognitive support provided by COGZ. The COGZ plugin has been included as part of the default Protégé installation since it was first developed (summer 2006).

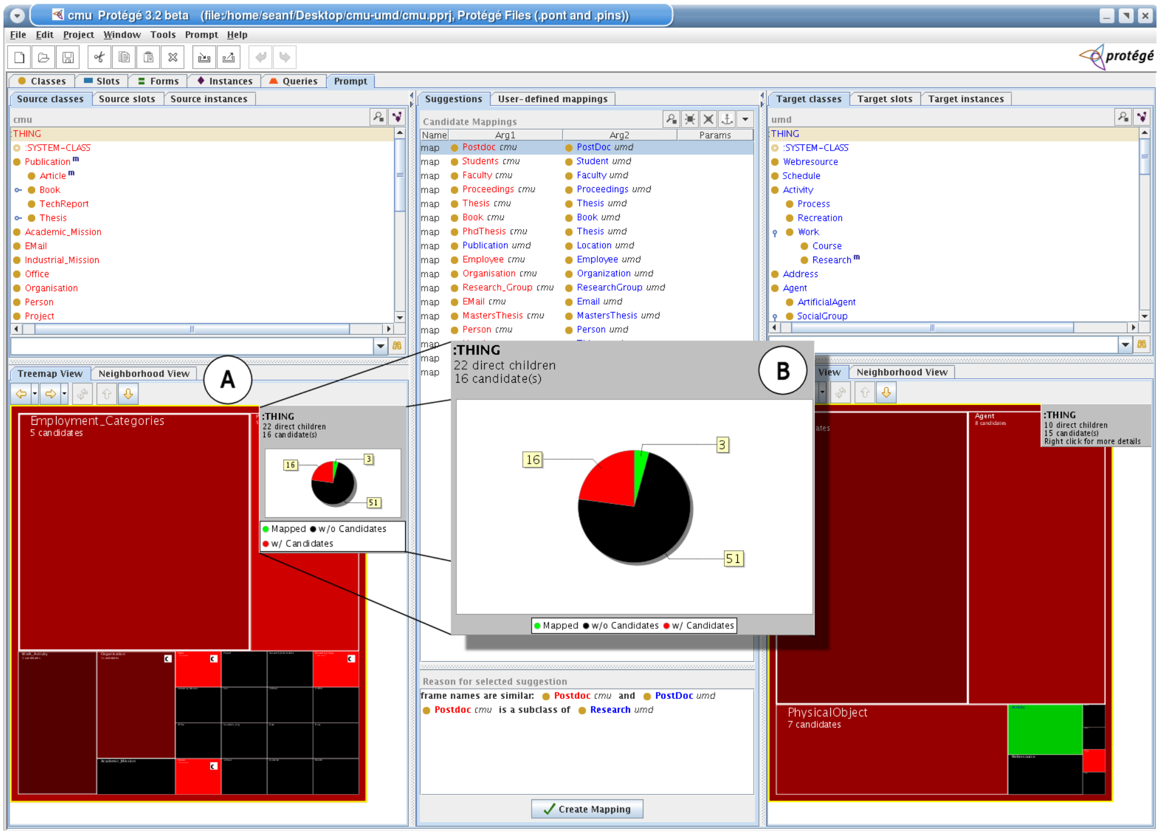

Figure 9.3. CoGZ *TreeMap view (A) with enhanced pie chart view (B). The color intensity corresponds to the number of candidate correspondences found within that region of the ontology. The pie chart provides an overview of how many terms have been mapped, have candidates, or have no associations within a region.*

9.2 Evolution of CoGZ

The CoGZ interface evolved as the framework emerged. The first version of CoGZ contained only a neighborhood graph visualization for helping users compare potential correspondences and a few options for filtering the suggestion list. In the second version we attempted to address most of the model requirements discussed in the previous chapter. This version included the original neighborhood comparison view from the first version, the suggestion list filters, and also a TreeMap [Shn92] interface for navigating the ontologies and highlighting the "candidate-heavy" regions of the ontologies (see Figure 9.3). A TreeMap is a 2-d space-filling representation of a tree, where nodes are represented as rectangles, and child nodes fill the space of their parent. The size of a rectangle is usually proportional to a metric chosen by the designer, like the size of a particular branch.

We originally chose this TreeMap-based interface for several reasons. The overview needed to fit a small area of the user interface yet display a large amount of data. Since TreeMaps are space-filling, they take up the same amount of screen regardless of ontology size. Also, since ontologies can be very large, we needed a visualization that scales well; TreeMaps can visualize several thousand nodes [FP02]. Color intensity in the TreeMap helps identify candidate-heavy regions of the ontology and mapped regions. The pie chart view provides details about the number of candidate correspondences, mapped concepts, and concepts without an association within each branch of the ontology. This gives an overview about what has and has not been completed within a branch of the ontology.

Both of these versions were integrated directly into the existing PROMPT UI. This put limitations on the type of support elements we could introduce. Based on our own experience using the second version, we recognized that adding features to PROMPT's existing UI was potentially introducing more complexity to the interface. Another problem was that with the plugin architecture we developed, we could easily add features, but it was difficult to take them away. Thus, although we could design our components as we chose, we could not manipulate the existing features of the PROMPT UI that we recognized as potential problem areas for users.

As a result, we extended the PROMPT plugin architecture to include support for *alternative perspectives*. This new extension allowed us to completely replace the default PROMPT UI, but still use PROMPT's other facilities. This provides a platform for allowing researchers to create, test and evaluate different ontology mapping UIs.

PROMPT's UI uses a list-based approach where all candidate correspondences are listed as a source term mapping to a target term. In the second version of COGZ, we explored using abstract representations of correspondences (e.g., TreeMap), but we determined that the learning curve was too great. In the observational case study (see Chapter 5), our users indicated that they liked the visual line-based representation of COMA++ where lines were used to indicate a potential correspondence by “linking” a source and target term. However, the users found the tool’s implementation very difficult to work with. We also anticipated that the line-based metaphor was potentially useful, but needed to be redesigned to reduce cognitive overhead. As a result, we designed the latest COGZ with a visual arc-based display for representing candidate and validated correspondences.

We believe this representation provides several advantages over a list-based display. For example, it is difficult to find all candidate correspondences associated with a given term by scanning a list. Every list item has to be checked for the term and one must remember what candidates for that term they have already inspected. Also, when inspecting a candidate correspondence, it is often important to inspect potential correspondences of the parent and sibling concepts. Since a list is not organized hierarchically, it does not facilitate this inspection process. This view, along with the filters discussed below, are the primary differences between this version and previous versions of COGZ. In the following, we highlight a variety of features of the COGZ tool and relate them to the cognitive support requirements that they address.

9.2.1 The main interface

Figure 9.4 presents the main COGZ interface. Figure 9.4(A) and 9.4(B) show the source and target ontologies (*Requirement 2.1*). Concepts with “C” icons represent terms with candidate correspondences that were discovered automatically, while concepts with “M” icons (e.g., Article) are terms that have been validated and mapped. Figure 9.4(C) shows a visual representation of correspondences. Candidate correspondences are represented by dotted, red arcs, while validated correspondences are represented by solid, black arcs. Figure 9.4(D) shows the main toolbar. Each ontology has a set of buttons for applying filters, moving through the correspondences, and representing the overall progress (*Requirement 4.6*). The central buttons correspond to global functionality for filtering and carrying out mapping operations. Finally, Figure 9.4(E) shows three tabs. The first tab displays all the candidate or suggested correspondences found automatically. The second tab displays

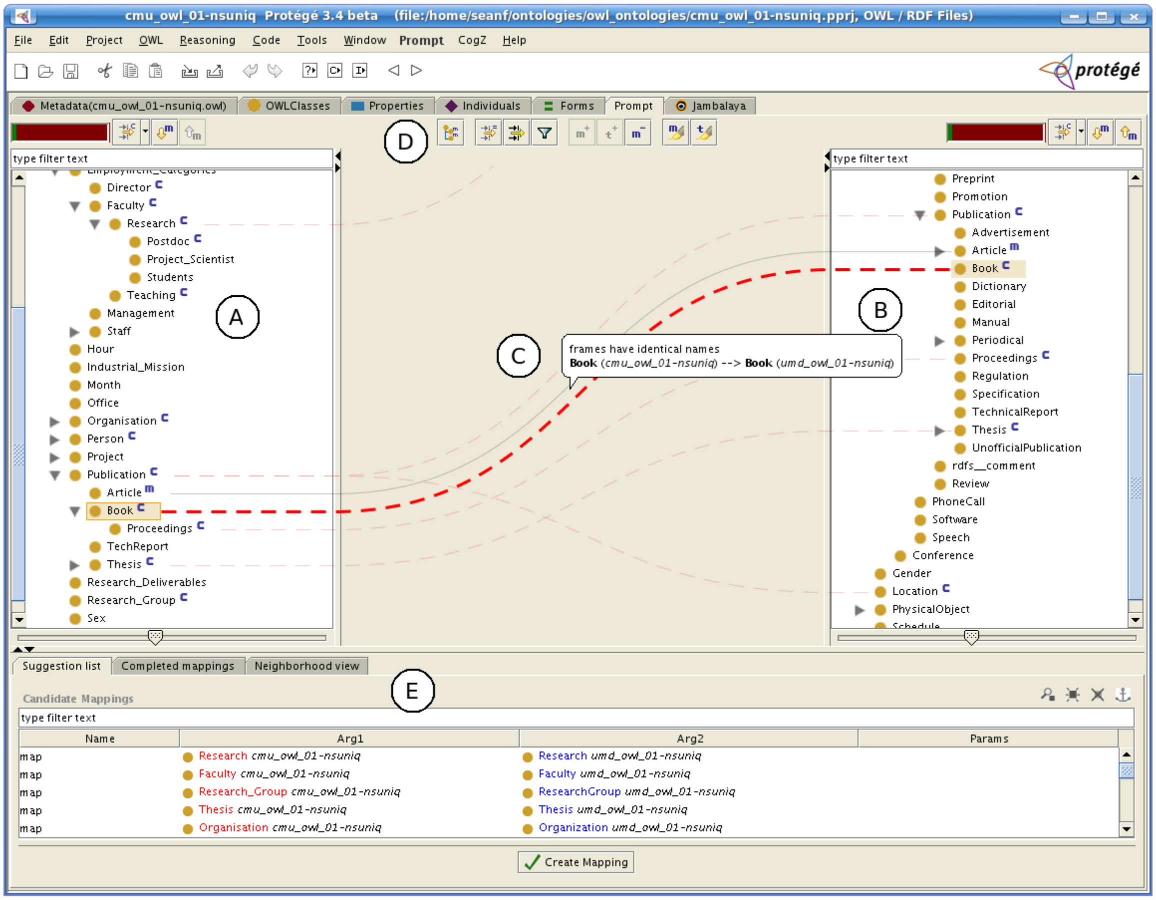

Figure 9.4. *The* COGZ *perspective in* PROMPT.

only the correspondences validated by the user. The final tab displays a side by side visual comparison between the concepts selected in the source and target ontologies (see Fig. 9.4).

9.2.2 Visualizing correspondences

COGZ uses a visual representation for candidate and verified correspondences (*Requirements 2.2, 2.3, 4.2, and 4.3*). Interaction with the visualization and ontology trees is synchronized. Selecting a node with correspondence candidates automatically expands the corresponding candidate terms in the other ontology and highlights the mapping arcs. This helps users explore all the candidates for a term. Multiple inheritance in an ontology is sup-

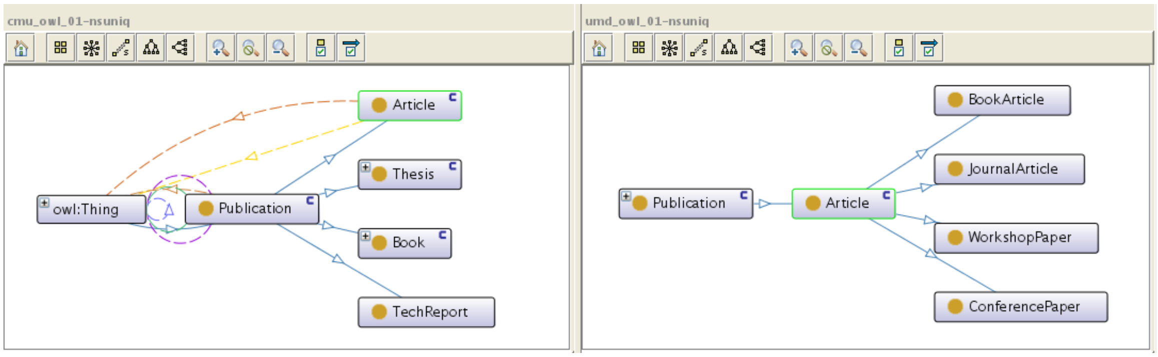

Figure 9.5. *The* CoGZ *neighbourhood view.*

ported by displaying alternative arcs travelling to the multiple hierarchies where a concept exists. The visualization helps support exploration and discovery of the parent, child and sibling correspondences of a term (*Requirement 1.1*), as clusters of potential correspondences are visible when two hierarchies closely correspond (see the Publication hierarchy in Fig. 9.4).

CoGZ supports the visual construction of user-defined correspondences (*Requirement 2.5*). A user can select a source concept and drag with the mouse to a concept in the target ontology. This will create a correspondence between the source and target term.

To help deal with some of the potential usability issues of such a display, we incorporated a variety of filters that helps users reduce the potential information overload that is inherent in this application domain. These are discussed in more detail in the "Filtering" subsection below.

Through a pilot study we determined that a list-based representation of candidate and validated correspondences was necessary to provide an alternative means of mapping navigation (*Requirements 2.2, 2.3, 4.2, and 4.3*). This helps users quickly inspect what was automatically found by the algorithm as well as what they have validated. The mapping visualization is synchronized with the list-based view and vice-versa. This synchronization helps reinforce learning the system and also helps support different styles of learning and exploration.

9.2.3 Neighbourhood view

The neighbourhood view provides a side by side structural comparison between candidate terms (see Fig. 9.5). Without this view, it is difficult to compare non-subclass relationships between terms, as these are not visible in the traditional hierarchical view of an ontology. Also, the neighbourhood view makes it easier to inspect multiple inheritance, which are not easily visible in a tree structure (*Requirement 1.3 and 1.4*). As we discussed in Chapter 3, humans tend to overuse categorical information for making inferences. Both this view and the mapping view help to clarify the categorical relations of concepts. We believe this facilitates the user's decision-making process.

In previous iterations of COGZ, the neighbourhood view was constructed by integrating the ontology visualization tool Jambalaya [SNM+02]. In the latest version, the dependency on Jambalaya has been removed and we have developed our own ontology visualization support. This was done in order to develop visualizations that corresponded more closely with our identified requirements. The neighbourhood view supports node and arc filtering based on ontological types, iterative navigation through continual expansion of nodes, a variety of different layouts, and zooming.

This component has been released also as a Protégé plugin called PQViz[1], for searching and visually navigating ontologies.

9.2.4 Filtering

COGZ supports a large variety of filters that have been based on the results of our previous user studies. These filters help support the user's mapping process as well as help users cope with the complexity of mapping ontologies. Search and filtering is critical as users can easily become overwhelmed when presented with large lists. Abrams *et al.* [ABC98] found that web browser users will not put more than 35 items in their favourites list before resorting to categorizing links within hierarchies or stopping their use of favourites

[1] http://webhome.cs.uvic.ca/~seanf/pqviz/pqviz.html

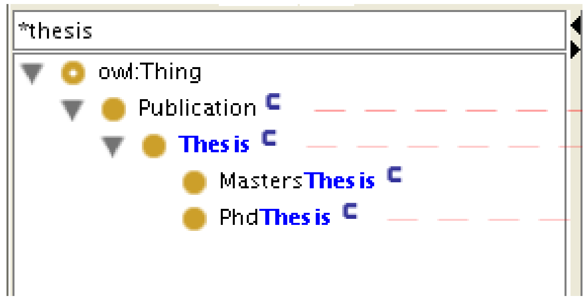

Figure 9.6. COGZ *ontology search.*

altogether.

The search for the source, target, candidate and verified correspondences is both a search and a filter for the corresponding display (*Requirement 2.4*). As a user types, the display is filtered to display only matching terms. In the case of the ontology searches, the parent hierarchy of matching terms is also included (see Figure 9.6).

The source and target ontologies can also be "re-rooted". That is, any term in the ontology can become the root of the tree. This removes all other terms that are not a subclass of the selected term from the display. It allows users to focus on particular areas of the ontology (*Requirement 4.5*). Also, the source and target ontologies can be filtered to display only the part of the ontology necessary to see the candidate or verified correspondences (*Requirement 4.4*). The mapping and ontologies can be filtered based on a selected correspondence as well. This will filter the display to show only the concepts necessary to inspect the currently selected correspondence.

Besides filtering, to help users cope with the potential clutter of this representation, mapping correspondences that do not have at least one visible concept in the display are automatically filtered from the view. Non-selected correspondence lines are semi-transparent and if the user desires, they can be filtered completely. All the filters are toggle buttons and can be used in combination, creating a powerful, user-driven approach to coping with potential complexity.

9.2.5 Reporting

Based on results from the survey and interview study, we incorporated several different reports into CogZ to facilitate discussions of mappings during team meetings. The first is a "Mapping Status Report". This is an HTML based report that gives a breakdown of the current mapping results, including each candidate and validated mapping (*Requirements 4.2, 4.3 and 4.6*). Candidate correspondences diplay the terms involved, the reason for the suggestion, and the type of match (e.g., exact, similar name). The second report, which is available for both the source and target ontologies, is a "Mapping Hierarchy Report". The report displays the terms that have been mapped, along with their parent hierarchy. This is essentially the implementation hierarchy of an ontology necessary to cover the terms involved in a mapping. This is potentially useful for large ontologies like SNOMED CT, where in most mapping scenarios, only a piece of the ontology is needed to cover the terms involved in the mapping.

9.2.6 Other features

We have improved Prompt's deletion support. To remove a validated correspondence from the native Prompt interface, the mapping storage ontology must be modified directly, which can result in a number of non-obvious steps. In CogZ, this is available as a single button action. To the user, there is no difference between removing a candidate item versus a fully mapped item (*Requirement 2.3*).

To help support the user's working memory, we introduced the notion of a "temporary mapping". This functionality allows a user to flag a correspondence as temporary, which are displayed in a different color and can then be filtered from the view (*Requirement 1.2*). These marked correspondences can be visualized on their own or with all other correspondences in the system. If a user performs a mapping operation with a concept that already has a temporary correspondence, the user is reminded about this and asked if they wish to proceed. If they proceed, the corresponding temporary correspondences are removed as possible candidate correspondences.

COGZ supports two different types of tree zooming (*Requirement 2.1*). Both the source and target ontology can be zoomed to display more or less information. Also, a "fish-eye" zoom is available that will display a selected term at the regular font-size and other terms progressively smaller as the terms get further away from the focal point. Multiple focal points can be selected. Zooming helps with exploration of large ontology hierarchies and multiple correspondences by allowing more information to be displayed in one view.

We have released the tree zooming component of COGZ as a separate open-source library [2].

9.2.7 Automation support

Requirements 3.1 through 3.4 all relate to the automated functions of an ontology mapping system. COGZ depends on PROMPT's native support of these requirements. However, the plugin framework we developed for PROMPT allows other developers to change how each of these requirements is addressed. Moreover, we demonstrate in Chapter 11 how we extended both COGZ and PROMPT to support an alternative means of generating and executing a mapping.

9.3 Summary

We described a plugin framework that we developed for the PROMPT ontology management suite. The framework helps address two fundamental issues. First, how can we satisfy the cognitive support requirements in one consistent environment, and second, how can we close the gap between mapping algorithm research and mapping users? By supporting user interface extension points in PROMPT, researchers interested in mapping interfaces can incorporate their ideas and tools to help support end users as well as incorporate their own domain or application specific requirements. Similarly, the algorithm extension points also

[2] http://sourceforge.net/projects/zoomablejtrees/

help the algorithm researcher. By using these extensions, researchers (or software developers) can easily incorporate their algorithms into PROMPT, allowing the research to be available under one consistent user interface. End users will benefit from having access to the best known algorithms, as well as the best cognitive support tools available.

We also described features of our mapping perspective, COGZ. The tool is based on the requirements outlined in the previous chapter. It combines a variety of visualization and filtering techniques to help address the opportunities of cognitive support derived from our user studies. In the next chapter, we evaluate the tool, first demonstrating the feasibility of the approach, second the effectiveness, efficiency, and accuracy of using the tool, and finally the adoption of both the tool and framework by researchers and industry.

Chapter 10

Evaluation

In this chapter, we address research objective **O8**, the evaluation of the tool and design. We begin by evaluating the feasibility of the visualization approach through a case study. We demonstrate that the approach is scalable to a large number of correspondences across two large anatomy ontologies. Following this, we evaluate the tool in a controlled lab experiment. The results from the experiment demonstrate that the COGZ perspective is preferred over the default PROMPT view, but more importantly, we explore why this is the case in terms of the cognitive support provided by the COGZ interface. Finally, we discuss the adoption of our framework and the COGZ tool by the research and industry communities.

10.1 Case study evaluation

In this section we discuss a mapping case study to demonstrate the feasibility, scalability and some of the features of the COGZ tool for use in the biomedical research domain. For the study, we chose to use two anatomy ontologies, one from the National Cancer Institute and the other from Jackson Labs. We chose these ontologies as they are both large, widely used, and well-known ontologies.

The National Cancer Institute (NCI) publishes the NCI Thesaurus monthly. The thesaurus is a reference terminology and biomedical ontology that is widely used by the NCI and by other academic institutions. It covers concepts from the cancer domain, anatomy, agents, drugs, chemicals and gene products. The terminology contains approximately 34,000 concepts, structured into 20 taxonomic trees [nci08].

The Mouse adult gross anatomy (MA) has been developed as part of the Gene Expression Database Project at Jackson Labs. It is a dictionary that organizes the anatomical structure for the postnatal mouse spatially and functionally using "is a" and "part of" relationships [Lab].

In 2006, the NCI and Jackson Labs participated in a project to map the human anatomy branch of the NCI Thesaurus to the MA ontology. The goal of the project was to switch the current mouse anatomy used by the NCI Thesaurus to the concepts stored in the MA ontology. Concepts from the human anatomy branch of the NCI Thesaurus were mapped to mouse anatomy concepts. Once the mapping was complete, the project members hoped to add the MA terms to the NCI Thesaurus and the mapping would be converted into relationships between the new terms and the existing terms.

The participants of the project originally explored using PROMPT to help carry out this process. However, they were not able to use this system as PROMPT relies heavily on computing lexical similarities between concept names, but both ontologies use unique ID values for concept names. For example, the concept "abdomen" in the MA ontology has the ID and concept name of "MA_0000029". The actual English name of a concept is stored in the *rdfs:label* attribute of a given concept. Due to this technical issue, the project members carried out the mapping manually, creating a list of 1545 correspondences, which they stored in an Excel spreadsheet. The spreadsheet consists of the MA ontology ID, *rdfs:label* value, NCI Thesaurus ID, and *rdfs:label* value.

The NCI was interested in using COGZ to explore these existing correspondences and possibly discover new ones. Several steps were required to make this possible as COGZ relies on PROMPT's algorithms, thus, the original problem the NCI experienced with using PROMPT would also exist in COGZ. Also, the MA ontology is developed using the OBO ontology format, but Protégé only reads OWL and Protégé frame ontologies. Finally, since the manually created mapping is only available in spreadsheet format, to be used in COGZ,

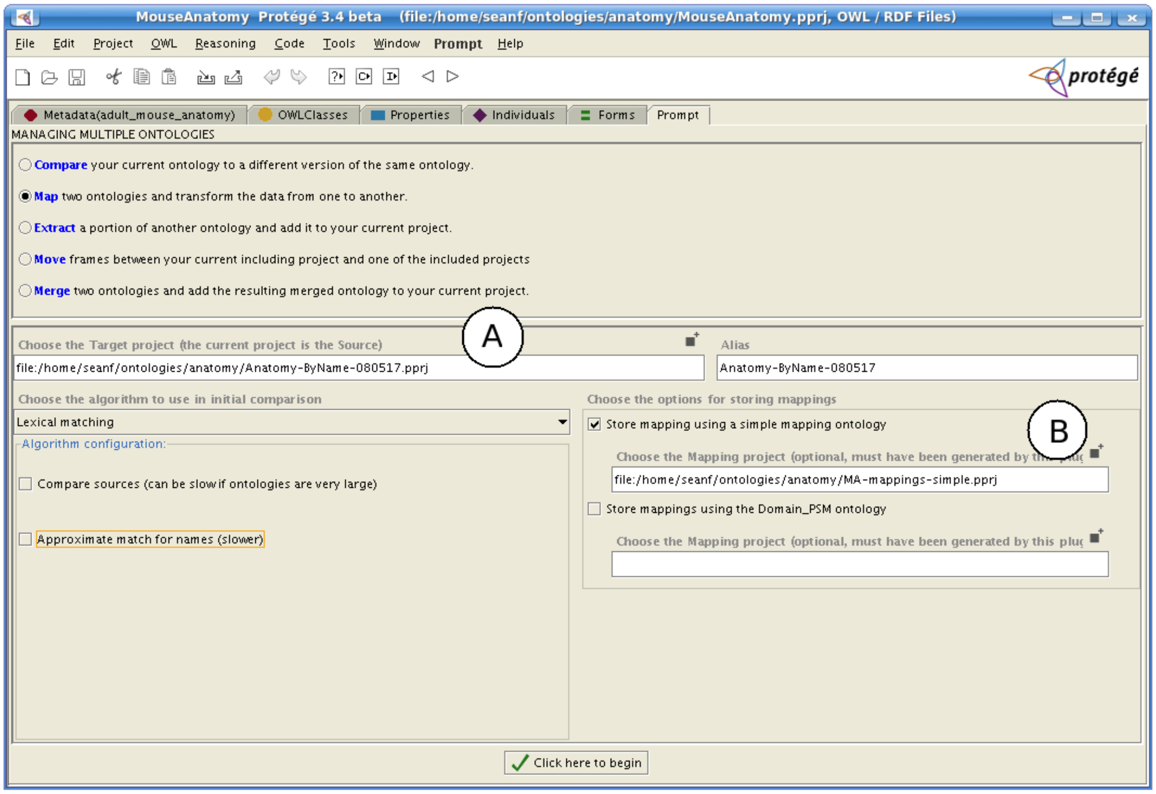

Figure 10.1. PROMPT *configuration screen for MA to NCI Thesaurus mapping. (A) shows the target ontology and (B) shows the existing mapping file.*

it would need to be converted into a mapping format that PROMPT supports.

10.1.1 Preparing the files

The first step towards supporting the NCI with using COGZ was to convert the existing MA ontology from OBO to OWL. As part of the NCBO project, a tool for converting between OBO and OWL has been developed [Wik]. The OBO converter is available as a Protégé plugin, and the conversion process is straightforward.

The next step was to convert the existing mapping into a PROMPT friendly format. To do this, we developed a small Java application that loads both ontologies using the Protégé API, reads the mapping from the Excel spreadsheet, and uses the PROMPT API to store the correspondences. The application reads each correspondence from the spreadsheet, and

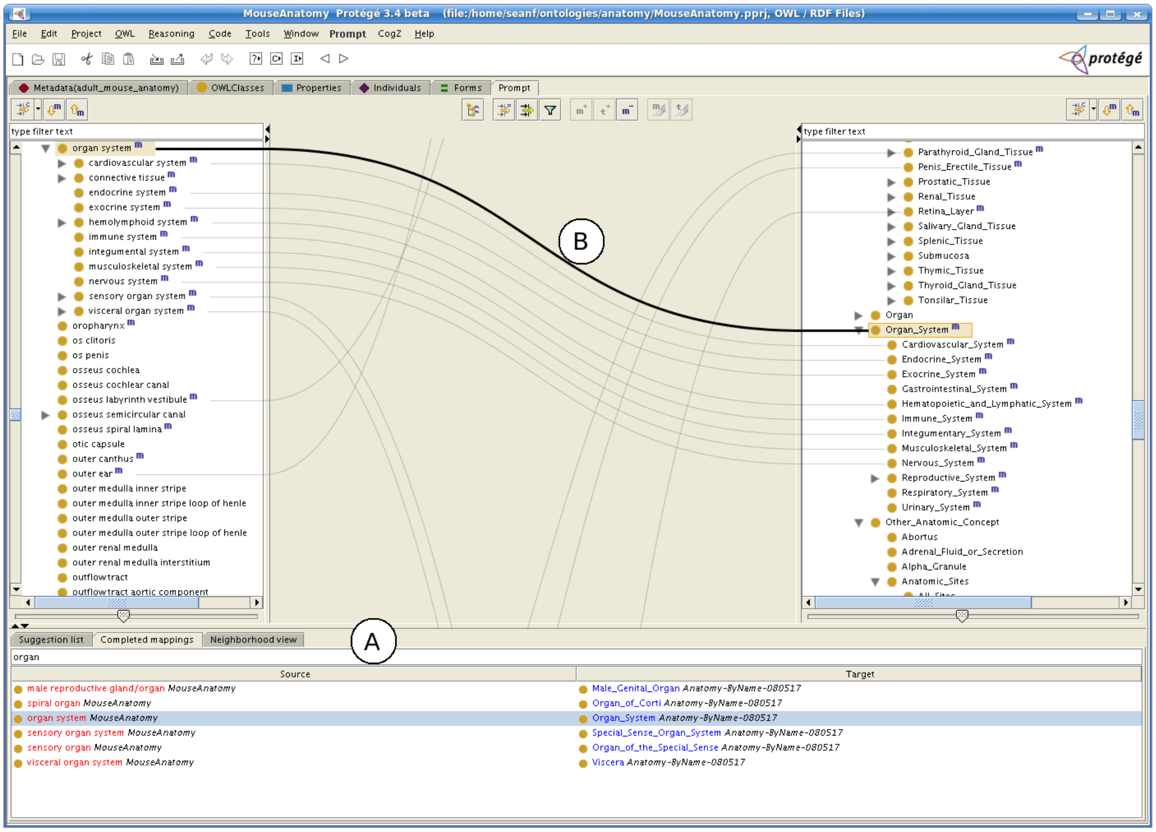

Figure 10.2. COGZ *showing the existing mapping between the MA ontology and NCI Thesaurus. (A) shows that the list of completed correspondences are filtered on the term "organ", while (B) shows the visual representation of the mapping from "organ system" to "Organ_System".*

uses the ID values to find the corresponding concept within the loaded ontologies. These concepts are then stored as "Class to Class" correspondences using the PROMPT mapping plugin architecture.

The final pre-processing step was to configure both ontologies to use the concept's *rdfs:label* as the concept name within Protégé. This process is supported by Protégé and described on the Protégé Wiki[1].

10.1.2 Loading the mapping

To explore the mapping using COGZ, the MA ontology is first opened in Protégé. The PROMPT plugin is then used to configure the mapping parameters. Figure 10.1(A) shows the target ontology, which is the NCI Thesaurus and Figure 10.1(B) shows the selection of the existing mapping ontology. The mapping algorithm configuration parameters are turned off as we do not want to generate candidate correspondences. Once the configuration is complete, the "Click here to begin" button is selected. In a separate thread, PROMPT loads the target ontology and mapping ontology and then the COGZ perspective. Figure 10.2 shows an example of what part of the mapping looks like.

10.1.3 Exploring the mapping

After the mapping has been loaded by PROMPT+COGZ, they can be explored interactively through COGZ's visualizations and filtering mechanisms. COGZ's visual representation makes certain mapping relationships easier to distinguish in a large mapping such as between the MA and NCI thesaurus. For example, Figure 10.3 shows that the Mouse Anatomy term "limb" has been mapped to two NCI Thesaurus terms. Although the trees are large, the information contained within them can be compressed through COGZ's fisheye zooming feature.

The mapping also help to highlight the modeling differences between the two ontologies. For instance, the NCI term "Cavity" is mapped to both MA terms "body cavity" and "body cavity/lining". The three subclasses of MA term "body cavity" have correspondences defined to subclasses of NCI term "Cavity". However, several subclasses of NCI term "Cavity" are mapped to MA terms outside of the MA "body cavity" hierarchy. Terms "Orbit", "Pelvis", "Abdominal_Cavity", and "Nasal_Cavity" are all mapped to MA terms that exist at the root level of the MA ontology. Consider Figure 10.4, which shows the highlighted correspondence between "nasal cavity" and "Nasal_Cavity". "Nasal_Cavity" is

[1] http://protegewiki.stanford.edu/index.php/HidingIdentifiers

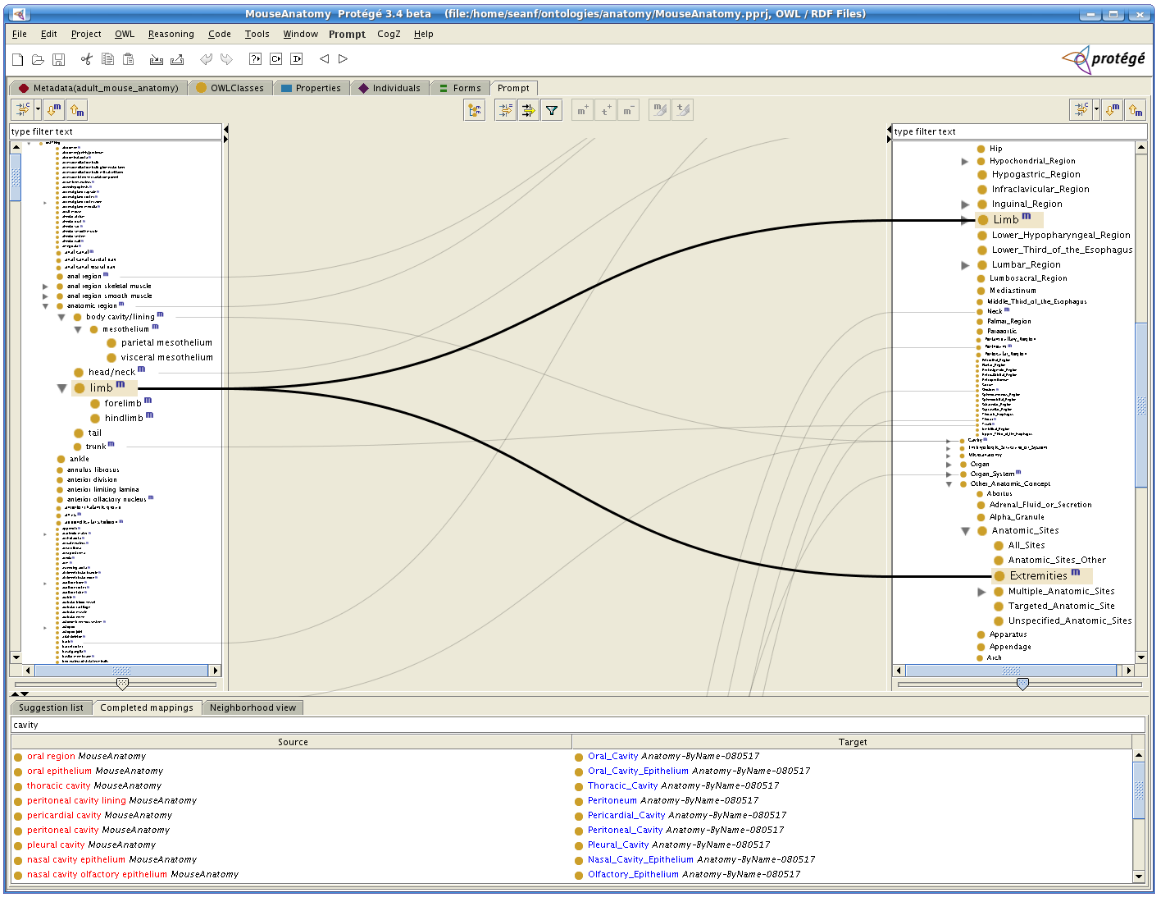

Figure 10.3. *COGZ showing two correspondences from the Mouse Anatomy term "limb" to the NCI Thesaurus terms "Limb" and "Extremities". The selected terms stand-out due to COGZ's fish-eye zoom feature.*

a child of "Cavity", which has correspondences in other locations of the MA ontology. The user can easily switch to viewing these correspondences by selecting the "Cavity" term. Without the support of the correspondence lines, tree representation of the class hierarchy, and synchronization between the two, it would be difficult to browse correspondences that exist far apart in one ontology but close within another.

10.1.4 Improvements to COGZ

Several improvements had to be made to COGZ in order to make it work efficiently with the two anatomy ontologies. Several technical improvements were made to COGZ due to

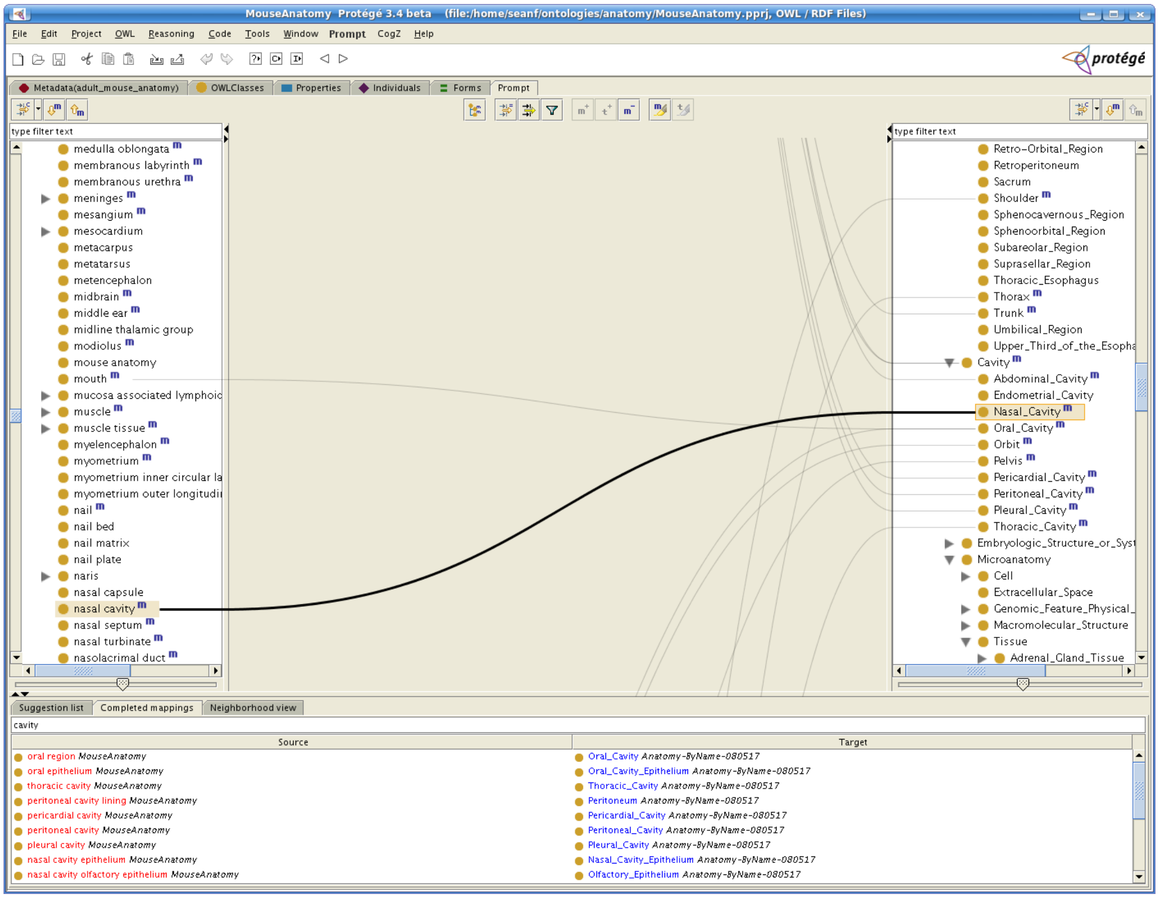

Figure 10.4. COGZ *showing correspondence between "nasal cavity" and "Nasal_Cavity". "Nasal_Cavity" is a child of "Cavity", which is mapped to two concepts in completely different locations than the MA term "nasal cavity".*

the size of the anatomy ontologies and the number of correspondences. Originally, COGZ did not scale well; the rendering of the mapping lines along with the ontology trees were too slow to be usable.

There were several reasons for this. COGZ tracks correspondences in memory based on the ontology conceptual structure. When rendering the mapping lines, COGZ converts the conceptual structure into the tree-based structure that represents the source and target ontologies. This conversion process is necessary as the tree structure is not static in COGZ. The filtering that is supported manipulates and changes the tree structure, so the structure and tree model may not match the conceptual model. However, it is an expensive operation to continually calculate the tree structure for every correspondence during rendering.

To avoid this, various caching mechanisms were put in place. Each time a tree path is constructed for the first time it is stored in a cache. Each time the tree model changes, the cache is cleared as the constructed tree paths are no longer valid.

Another expensive operation is constructing and tracking multiple inheritance in the conceptual model. Since an ontology concept may have multiple paths to the root concept, correspondences must be created to represent all possible paths. To avoid excessive computation, when a mapping is loaded by COGZ, these object paths are calculated and cached.

To avoid excessive line rendering, only mapping correspondence lines with at least one concept currently visible are displayed. This avoids long lines from concepts that are offscreen polluting the mapping interface. A similar approach is used by Microsoft's BizTalk schema mapping interface [RCC05b].

10.1.5 Moving forward

The NCI is interested in using COGZ to update their existing mapping. The thesaurus has changed since the mapping was first created and since their original mapping process was largely manual, it is difficult to keep up to date. COGZ provides several advantages to exploring correspondences over a simple spreadsheet. For example, both ontologies and correspondences can be searched interactively, the hierarchical structure of the terms is visible, correspondences with multiple inheritance are displayed visually, and new correspondences can be easily created. The NCI is hoping to use COGZ for their next mapping update. We are working with them to incorporate feature suggestions such as being able to verify multiple suggestions simultaneously and persist temporary correspondences. We are also making improvements to PROMPT's loading efficiency. Also, the NCBO used the PROMPT version of the mapping that were constructed for this study in their BioPortal application.

10.2 User study

In this section we discuss a user study we conducted to help evaluate COGZ. This evaluation is complementary to the previously discussed case study. The case study helped to validate the tool from a technical point of view, and allowed us to experiment and make adjustments to our approach. The user study is focused on the *cognitive support* provided by the tool. Measuring this effect directly is difficult. As discussed in Chapter 4, to measure cognitive support we must attempt to assess the computational benefit that a tool provides. In this study, we propose to measure this effect by evaluating how effective, efficient, and satisfying the tool is for the user.

COGZ has been included as part of the default Protégé download since 2006. However, no formal evaluation of the tool had been conducted. In agreement with Noy and Musen [NM02], we believe that larger and more comprehensive user evaluation studies need to be carried out in this area.

We limited the scope of the study to research questions that addressed the first four information needs discussed in Chapter 8, Section 8.1. Our primary focus was on evaluating the cognitive support COGZ provided through its visual mapping interface, its visual neighborhood view, and its filtering mechanisms. Our research questions are as follows:

Q1: Does an arc-based visual representation of correspondences provide more cognitive support than a list-based representation?

Q2: Does a visual graph-based representation of a concept's local structure (e.g., parents, siblings, children) provide more cognitive support than a form-based representation?

Q3: Do mapping/ontology filters help reduce task complexity and aid in the completion of mapping tasks?

10.2.1 Hypothesis generation

Based on our research questions, we formulated several hypotheses that we wanted to evaluate. The study was designed to provide data on each of these hypotheses:

H1: Using COGZ, users will be able to complete complex tasks with more accuracy and faster than with PROMPT.

H2: Users will prefer COGZ and find it easier to use than PROMPT.

H3: Users will be able to understand a concept's structure more accurately with COGZ than with PROMPT.

H4: Users will determine that COGZ is more effective than PROMPT at supporting their decision-making process.

H5: Users will perceive that the COGZ filters help reduce complexity.

10.2.2 Method

10.2.2.1 Participants

We recruited 18 participants (five female, 13 male) for the study, all with computer science backgrounds (two undergraduates, 14 graduates, and two post-graduates). Participants were recruited via mailing lists and postings within the University of Victoria Engineering Computer Science building (see Appendix J). No specific knowledge of ontologies or ontology mapping was specified as a requirement. Ages ranged from early 20s to early 40s and user experience with ontologies and mapping ranged from completely inexperienced (nine participants) to expert (two participants). Participants considered to be inexperienced or "novice" were those that did not know what an ontology was and had no experience with mapping. Experts were those that worked with ontologies on a regular basis and had experience with mapping. Participants falling in between the two categories were considered to have a medium level of expertise and were those that knew what an ontology was, but did not work with ontologies on a regular basis.

10.2.2.2 Materials

For each tool, nine tasks (18 total) were assigned. These tasks consisted of five mapping and four comprehension level tasks. These tasks were based on the type of activities ob-

Table 10.1. *Example of study tasks*

1a. Complete the one-to-one mapping for concept **Attacker** from the source ontology if any exist.
2a. Complete the one-to-one mapping for concept **CornerArea** from the source ontology if any exist.
3a. Complete the one-to-one mapping for concept **Applause** from the source ontology if any exist.
4a. Complete all mappings in the branch **OtherFootballPlayer** of the source ontology. This includes this concept along with any child concepts. Remove any invalid suggestions.
5a. Map concept **FootballTimeDuration** from the source ontology to **Duration** in the target ontology.
6a. Should the concepts **Obstruction** from the source ontology and the concept **Do_Obstruction** from the target ontology be mapped? Why or why not?
7a. The tool suggests that **PlayOff** from the source ontology and **Player** from the target ontology should be mapped. Why does the tool make this suggestion?
8a. What children does concept **FootballPlayer** from the source ontology have? What parents does it have? What other (non-mapping) relationships exist for this concept?
9a. How many concepts from the source and target ontologies were mapped during this session?

served and recorded in our previous studies. To construct the tasks, we first classified the known mapping between the ontologies based on the categories of correspondences that the PROMPT algorithm produces such as exact match, lexical similarity match, and synonym match. We then randomly selected the concepts for our tasks from these categories. We did this to reduce any bias for one tool but guarantee that we would have tasks from each mapping category. For each task, there were two variations (an a and b version). We limited our focus to correspondences due to equivalence or a mapping to a more general concept.

The source and target ontologies used for the experiment were two OWL [2] ontologies describing soccer [3]. The source consisted of 439 concepts and 876 relationships. The target consisted of 275 concepts with 437 relationships. The mapping algorithm suggested 239 candidate correspondences. Although these are modestly sized ontologies, they have a large degree of multiple inheritance and both ontologies are further complicated by depending on concepts from the upper ontologies DOLCE [4] and SUMO [5].

An example of the study tasks are shown in Table 10.1. The first three tasks test differ-

[2]http://www.w3.org/TR/owl-features/
[3]http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/ontologies.htm
[4]http://www.loa-cnr.it/DOLCE.html
[5]http://www.ontologyportal.org/

ent variations of creating a simple correspondence. Concept **Attacker** has an exact match in the target ontology and this is detected automatically. Concept **CornerArea** has a match based on partial lexical similarity and is found automatically and also has an invalid suggestion. Concept **Applause** does not have an automatic correspondence and the proper correspondence is to a synonymous term.

10.2.2.3 Procedure

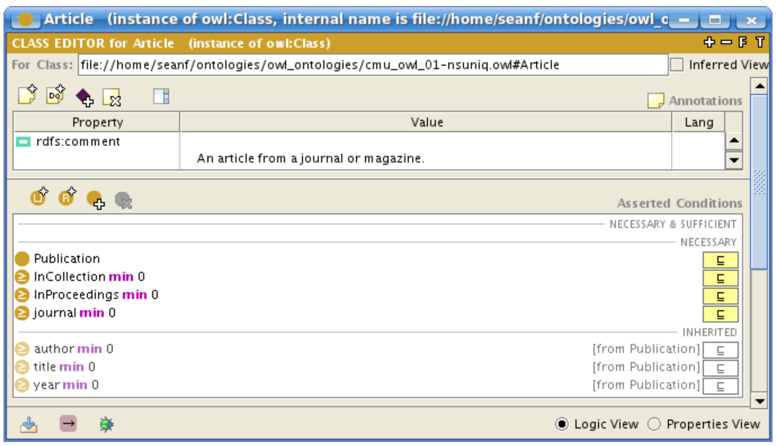

Figure 10.5. *PROMPT's form-based relationship view for the concept "Article". The view displays of the concept and "Asserted Conditions", which specify the parent of the concepts, direct domain-range constraints, and inherited constraints.*

We conducted a controlled lab experiment where participants carried out tasks with COGZ and PROMPT. PROMPT supports a list-based representation of correspondences and form-based representations of relationships (see Figures 9.2 and 10.5), which matched well with our research objectives. Since COGZ only replaces the user interface of PROMPT, both systems used identical algorithms and back-end software.

Nine participants solved nine tasks using PROMPT first and then solved the remaining nine tasks with COGZ. This order was reversed for the remaining nine participants. For each question, participants were randomly assigned to either version *a* or version *b*, and no participant solved the same problem twice. For example those that solved task *1a* with

PROMPT, solved task *1b* with COGZ. This random selection helped mitigate learning effect between tasks and tools.

The experiment followed the procedure described by the "Experimenter's Handbook" in Appendices L and M. Once the consent form was signed (see Appendix K), the first part of the experiment began with participants receiving training with the first assigned tool. They were instructed about ontologies, mapping, the tool's user interface, and finally asked to carry out several mapping tasks between two small domain related ontologies. After training, the experiment ontologies were loaded.

Once the ontologies were loaded, the participant was given a set of nine randomly selected tasks. The participants were instructed to "think aloud" [ES80] during the completion of the tasks. Once the tasks were completed, the participant was given the System Usability Scale (SUS) questionnaire [Bro96] to complete (see Appendix N). SUS is a standardized usability test that covers usability aspects of the system, such as the ease of use, the need for support, and complexity. This questionnaire was chosen because it is known to yield reliable results with reasonably small numbers of users [TS04].

After completion of the questionnaire and a short break, the same procedure was followed for the second tool (including training). The study concluded with a short interview as described in the Section 10.2.2.4.

10.2.2.4 Data collection

All studies were audio and video recorded, and notes were taken during the sessions. During the task completion stage of the study, the time to complete a task and task accuracy were recorded. We scored participants with an accuracy of 0 on a task that they failed, 1 if they successfully completed 50% or more of the task, and 2 if they completed the task correctly. The mapping results were saved for each tool during each session for later inspection. The SUS scores were evaluated and recorded after each study. All interviews were transcribed for later analysis. The questions asked during the interview were as follows:

1. Which tool did you prefer? Why?

2. What did you dislike about either tool?

3. Which tool did you feel more confident that your produced correspondences were correct?

4. Can you describe your process for how you verified a correspondence?

5. Which tool better supported this process? Why?

10.2.2.5 Analysis

We analyzed the quantitative data to test our hypotheses and, in turn, answer our research questions. We compared means for task times and accuracies for each task, overall accuracy, overall SUS scores, as well as each SUS response via a paired, two-tailed t-Test. For all tests we used $\alpha = 0.05$. For comparing the means of individual task times, we used results only from the experiments where the participants were able to successfully complete at least 50% of the task. Also, we eliminated outliers from the timing data that were more than two standard deviations away from the mean.

We also analyzed the qualitative data to help reveal why one tool was preferred over the other. For the interview data, we conducted a preliminary exploratory analysis [Cre03], followed by a coding process that involved segmenting and labelling the text from the interviews (see Appendix O). Coding was performed by two researchers to improve our interpretation. From the coding, a number of themes emerged.

10.2.3 Results

We first present our quantitative results and verify our hypotheses. Our hypotheses help address our research questions and directly compare COGZ to PROMPT. Following this, we discuss results from our qualitative analysis, which helps explain why one tool was preferred over the other.

Table 10.2. *Overall comparison results*

Test	Mean (CogZ)	Mean (Prompt)	P-value
Overall Accuracy	1.69	1.58	$0.01, t(17) = 13.04$
Overall Time	18:27 mins	21:40 mins	$0.04, t(17) = 5.04$
SUS	73	46	$< 0.001, t(17) = 98.95$

10.2.3.1 Quantitative results

Our first research question asks whether a visual representation of correspondences provides more cognitive support than a list-based representation. To determine this, we test our first two hypotheses: **H1:** *using* CogZ*, users will be able to complete complex tasks with more accuracy and faster than with* Prompt, and **H2:** *users will prefer* CogZ *and find it easier to use than* Prompt..

To test the first hypothesis we computed paired, two-tailed t-Tests, comparing individual task times and accuracies between CogZ and Prompt. We first compared the overall mean accuracy for all tasks. As shown in Table 10.2, CogZ had a mean of 1.69, while Prompt's mean was 1.58 ($t(17) = 13.04$, P-value $= 0.01$). We eliminated the three tasks where all participants performed perfectly in order to reduce the ceiling effect of everyone scoring 2 on accuracy. In this test, CogZ had a mean value of 1.57 and Prompt 1.37 ($t(17) = 21.43$, P-value < 0.001). Thus, users performed more of the tasks correctly with CogZ than Prompt.

We performed a similar set of calculations for comparing the overall task times, see Table 10.2. CogZ's mean to complete all tasks was 18 minutes and 27 seconds, while Prompt's was 21 minutes and 40 seconds ($t(17) = 5.04$, P-value $= 0.04$).

Prior to the study, we considered tasks $4a$ and $4b$ (the branch mapping tasks) to be two of the most complex tasks, as they involve constructing several correspondences via suggestions and manual construction, and removing invalid suggestions. There was no statistically significant difference for the time to complete $4b$. However, this task involved fewer steps than $4a$. Thus, it may not have been complex enough to create a divergence in task time performance. The only task with statistical significance when comparing task completion time was task $4a$. In this task, on average COGZ users took 7 minutes and 39 seconds while PROMPT users took 11 minutes and 40 seconds ($t(17) = 2.94$, P-value = 0.020).

We performed similar calculations for comparing individual task performance accuracies. The results with statistical significance are summarized in Table 10.3. We see that with tasks $4b$, $8a$, and $8b$, on average, COGZ users perform over 50% of each task correctly, while PROMPT users perform less than 50% correctly. It is also interesting that although there was no calculable difference in task time performance for task $2b$, there was a significant difference in how accurately that task was performed between the two tools.

To test our second hypothesis, we compared the average SUS scores between the tools. Scores in SUS range from 0 to 100, a higher score being better. COGZ had an average score of 73, while PROMPT scored 46 ($t(17) = 98.95$, P-value < 0.001) (Table 10.2). Also, all of our users indicated in the interview session that they preferred COGZ over PROMPT.

Our second research question asks whether a graph-based representation of a concept's local structure provides more cognitive support than a form-based representation. To answer this, we evaluated hypotheses: **H3:** *users will be able to understand a concept's structure more accurately with* COGZ *than with* PROMPT, and **H4:** *users will determine that* COGZ *is more effective than* PROMPT *at supporting their decision-making process*. Tasks $8a$ and $8b$ specifically asked participants to determine the structure of a particular concept. As indicated in Table 10.3, participants were able to more accurately describe the full structure of a particular concept using COGZ when compared to PROMPT. Our final interview question helped us evaluate **H4**. Seventeen of our 18 participants indicated that COGZ better supported their decision-making process.

The final research question asks whether filtering helps reduce task complexity. To answer this, we tested hypothesis **H5:** *users will perceive that the* COGZ *filters help reduce complexity*. This is a difficult hypothesis to test as we could not directly measure a user's perception and mental effort. However, we did observe that users relied heavily on the filters while using COGZ. The primary filter used was COGZ's search and filter feature on

Table 10.3. *Task accuracy comparison*

Task	Mean (CogZ)	Mean (Prompt)	P-value
$4b$	1.57	0.91	0.020, $t(17) = 3.00$
$8a$	1.44	0.67	0.020, $t(17) = 2.93$
$8b$	1.56	0.67	0.020, $t(17) = 2.88$

the ontology trees and suggestion list along with the concept re-rooting feature. The search filter was used by all participants and users primarily had positive reactions to this feature (see Section 10.2.3.2). Two users made use of the candidate mapping filter, which filters a given ontology to display only those concepts with candidate correspondences along with their parent structure. This feature was not presented during the training session, however these users discovered it by exploring the buttons in the CogZ toolbar. One user also used the background mapping line filter to help reduce the complexity of the display, and again, this feature was not presented during training.

10.2.3.2 Findings

In this section we explore why CogZ outperformed Prompt. This exploratory analysis is based on the results derived from our interview and "think aloud" data. From both of these sources we discovered several themes. For each of these themes, we provide illustrative quotes from the participants.

Local structure aids decision-making. All of our participants indicated that inspecting the parent, sibling, and sometimes children of a concept were essential in helping them decide whether a correspondence was valid or not. Of course, the name of the concepts was also key, but matching parents or similar siblings reinforced their confidence about the match even when the terms were unfamiliar. In the branch mapping tasks, several participants, despite limited domain expertise, were able to infer that the soccer terms "Defense" and "Back" conceptually meant the same thing by relying solely on matching local structure. The neighborhood view also facilitates the inspection of this information. *"In CogZ,*

. . . you could very easily see the neighbors of the source and the neighbors of the target and if those were matching up then that would suggest that ok, this is much more likely to match than some other ones."

Visualization coincides with the user's mental model. The arc-based mapping visualization in COGZ helped support the user's mental model for how they understand and interpret a conceptual correspondence. Many of our participants indicated this as one of the main reasons they preferred COGZ. For example, participants stated that, *"COGZ points out mappings."*, *"When I imagine relationships, this is exactly what I see."*, and *"It sort of just allowed me to see if the mappings matched my mental model."* The match between the visualization and the user's mental model is key to reducing the overload involved with actually evaluating and understanding what is produced by the algorithm.

The visualization utilizes the principle of perceptual substitution. By visualizing a potential correspondence as a direct, but temporary link, users are able to rely on their fast perceptual operations for immediately observing and understanding the structure and semantics of that match. However, with a purely list-based approach, users need to actually find and read a suggestion, then exert mental effort to understand how those concepts and their local neighborhoods align. As one user put it, *"To have no support for something that is a visual process like tree comparison makes it really difficult."*

Filtering supports the mapping process. As mentioned previously, COGZ's search based filtering and concept re-rooting were relied on heavily by the participants. Generally, when participants knew the term they wished to map in the source ontology, they would search the source, which in COGZ would limit the visible suggestions to that term's local neighborhood. All possible candidate correspondences were visible for inspection. One of the participants commented, *"With filters I can quickly see what parents it has and then to be able to see all the candidate correspondences for that region quickly was key because then I could easily see that some were right and some were wrong and start cleaning up that* *sub-display."*

This feature is an example of the cognitive support principle ends-mean reification. That is, filtering and displaying just the focused concept's candidate correspondences gives the user a concrete means to progress towards their goal of mapping the given concept. They know they must work through each link and verify its validity. The solution will be a concrete link connecting a source and target concept or all links will be removed because the concept has no valid correspondence.

Conversely, in PROMPT this process was not well supported. Searching an ontology was not linked with evaluating candidate correspondences. There was no direct correlation between the tree's display of a concept and the suggested correspondences available for that concept. Participants would first search the source and then have to scan the list of suggestions looking for ones involving their term of interest. As a result, many of the PROMPT users actually ignored suggestions: *"I found I didn't use the suggestions in Prompt."*, and *"I found I wasn't really using the suggested relationships in Prompt because they were hard to use."*

Arc-based display aids discovery of correspondences. The visual mapping display in COGZ enabled users to discover new correspondences. For example, if a term needed to be mapped but did not have any suggestions, participants would often look at suggestions for other terms in the local neighborhood. Valid suggestions in the local neighborhood of a term gave the participants a hint about where to focus their effort for discovering a new correspondence. Several participants explicitly commented about this during the interview: *"The fact that the mappings of an entire hierarchy were visible say when you were looking at a child you could see its sibling recommended mappings and also see its parent mappings."*, and *"With CogZ I could just see the mappings and it helped me navigate by highlighting similar regions of the ontology."* This clustering effect also reinforced the participants' confidence about the matches they were evaluating because if many sibling terms have matches in the same area of the target ontology, the candidate match being evaluated is more likely to be valid. Thirteen of the 18 participants (two undecided) indicated they were more confident about the matches they produced while using CogZ.

Visualization reinforces understanding of an ontology. The neighborhood view and mapping visualization helped reinforce user understanding of the source and target ontologies. The neighborhood view provided a quick way to visually inspect and compare the complete local structure of a term. One participant noted that, *"Only through the visualization could I see other relationships. I didn't know how to do that with Prompt."* As the discussion of hypothesis **H3** indicated, users performed better on the conceptual task of actually understanding and describing the semantics of a given concept.

This visualization is an example of the cognitive support principles of redistribution and perceptual substitution. Without the neighborhood view, the user needs to reconstruct in their "head" a comparison between the contexts (e.g., parent, siblings, children, and properties) of the concepts of interest. In CogZ, this effort is redistributed to the tool where the visualization provides this information in a side by side comparison. The user can then rely on their fast visual processing to compare the two concepts.

Filtering and visualization facilitates focusing on complex issues. Due to better support in CogZ for verifying candidate correspondences as well as carrying out user-defined correspondences, users were able to spend more of their time addressing complex mapping issues, such as one-to-many relationships, correspondences from specific to more general concepts, and matches based on synonyms rather than lexical similarity. Reducing the effort involved with carrying out simple operations is important as a large majority of correspondences are simple one-to-one exact or similar name matches. It is the difficult correspondences, requiring domain expertise and possibly multi-person discussion, that requires human involvement.

Every tool implicitly encodes a process and it is important that the supported process matches the user's work process. Participants were frustrated by the mismatch between their own process and the way PROMPT supported mapping. For example, *"I think what it boils down to is when I was using Prompt I started getting pretty frustrated because it wasn't really helping me at all, it was almost being somewhat cumbersome, whereas CogZ let me do things quickly and easily."*, and *"I found I was never quite sure where to do mappings in Prompt. Do I have to find it in the suggestion list or do I have to do it myself? It was just a bit more confusing."*

10.2.4 Limitations

Due to our sample size and the relatively small number of subjects able to complete the more challenging tasks, we were not able to compute statistically meaningful timing comparisons for all tasks. For simpler tasks, there was little difference between the tools, however, a larger sampling may reveal trends not yet detected. Also, it may have been beneficial to analyze the difference in performance across populations, such as expert versus novice users. However, experts are difficult to recruit for such a study and thus only a small number were available. Our tasks may have also been biased towards participants performing better using COGZ. We tried to address this by first randomly selecting the concepts for our tasks from a list of candidate concepts. We also randomized which tasks were assigned to users and tools, and based the tasks on real mapping scenarios.

Another possible limitation was that participants may have been able to infer that COGZ was the "preferred tool" since it had more features. We attempted to mitigate this by randomizing the order the tools were used and by having the SUS questionnaires answered immediately after each tool was used. Finally, we believe our experimental design could be improved by incorporating more complex mapping tasks that test specific types of mapping scenarios, such as one-to-many correspondences and granularity differences. In the future, we plan to carry out more studies to address these issues.

10.3 Adoption

In this section we briefly discuss some of the adoption of CogZ and the mapping framework. We discuss adoption in order to help provide context for how different researchers have found this work useful.

Both the CogZ tool and framework have been adopted by a number of researchers and non-academics working in industry. It is difficult to know the exact number of CogZ users. The tool is available with every Protégé download, but a download of Protégé does not necessarily imply the user will be using CogZ. However, on the CogZ website[6] alone, there has been an average of approximately 10-20 downloads per month over the last year.

The Prompt plugin framework that was originally developed as part of this work to support extensions to Prompt has also been adopted by a number of researchers. There exists four different algorithm plugins that come with Prompt's default installation and a number of other researchers have enquired about using the framework.

CogZ is one of the systems studied in the "Semi-Automatic Information and Knowledge Systems" course taught at the Vienna University of Technology. Students must write six to eight page papers that include describing how it works and what are the benefits of the system. Also, the first paper that introduced CogZ (see [FNS06]) is one of two research papers the students may also study.

Both Gennari *et al.* [GNCC08] and Gil *et al.* [GBRC08] have used CogZ's neighborhood view for visual validation of correspondences. Recently, Meilicke *et al.* [MST08] helped motivate their work on reasoning support for manual mapping revisions by discussing the conflict resolution and inconsistency checking requirement highlighted by our framework.

In industry, the design principles introduced in Chapter 8, have been adopted by Pragati Synergetic Research Inc.[7]. I was hired by Pragati to use the design principles to redesign their user interface in their data mapping and analysis tool Expozé. Also, within industry,

[6] http://webhome.cs.uvic.ca/~seanf/cogz/cogz.html
[7] http://www.pragati-inc.com/

the Lilly Singapore Centre for Drug Discovery has been evaluating COGZ for possible use.

As mentioned previously, the NCI is interested in using COGZ to update their existing mapping. Also, NCBO, which in part has funded this project, is interested in incorporating a web-based version into their tool BioPortal (see Chapter 11).

10.4 Summary

In this chapter we evaluated different aspects of the COGZ tool and framework. We first evaluated the scalability of the approach by demonstrating how to visually explore a mapping between two large anatomy ontologies. As a result, both the NCI and NCBO are interested in adopting parts of COGZ to help them perform future mappings. We also presented the design and results from a controlled lab experiment. We believe the design of this study can be used as a template for other researchers interested in evaluating the user support their tool provides for ontology mapping. Moreover, the alternative perspective feature we introduced into PROMPT gives researchers a unique opportunity to use an identical back-end process during the experiment so that the focus is on the user interface.

We also presented implications and findings of our experiment that help to explain why users preferred COGZ and were able to more competently complete the study tasks. These findings demonstrate the importance of incorporating elements of cognitive support into the mapping process to assist users with constructing an ontology mapping. We must harness the symbiotic relationship between human processes and the tool's automated process to allow users to work more efficiently and effectively.

Finally, we discussed some of the adoption that has taken place since COGZ was first introduced. In the next chapter, we discuss extending COGZ, specifically adapting the system to perform data transformations and also visualize mappings on the web.

Chapter 11

Extending CogZ

In this chapter, we discuss extensions to CogZ. We first discuss how we are adapting the CogZ visualization to the web for integration into the NCBO's BioPortal application. Following this, we discuss how we adapted the CogZ tool to perform model mapping and data translation between software models. We use this technique to develop visualizations for ontology instance data. Part of this work was originally presented in an earlier paper [FBGS09].

11.1 Web-based mapping visualization

In this project, we are adapting the CogZ approach for exploring and navigating correspondences to the web. Eventually, this web-based version will be integrated into BioPortal to assist users with exploring an existing mapping between two ontologies. Currently, BioPortal only displays mappings as a list with links to the relevant concepts.

The web-based CogZ is being developed using Adobe Flex[1], which provides a powerful platform for developing web-based applications. We have had previous success with developing graph-based visualizations for BioPortal using this technology. Communication with BioPortal occurs through web services, where the services provide access to the ontologies, concepts, and mappings stored in the repository.

All communication happens asynchronously and on-demand based on the user's inter-

[1] http://www.adobe.com/products/flex/

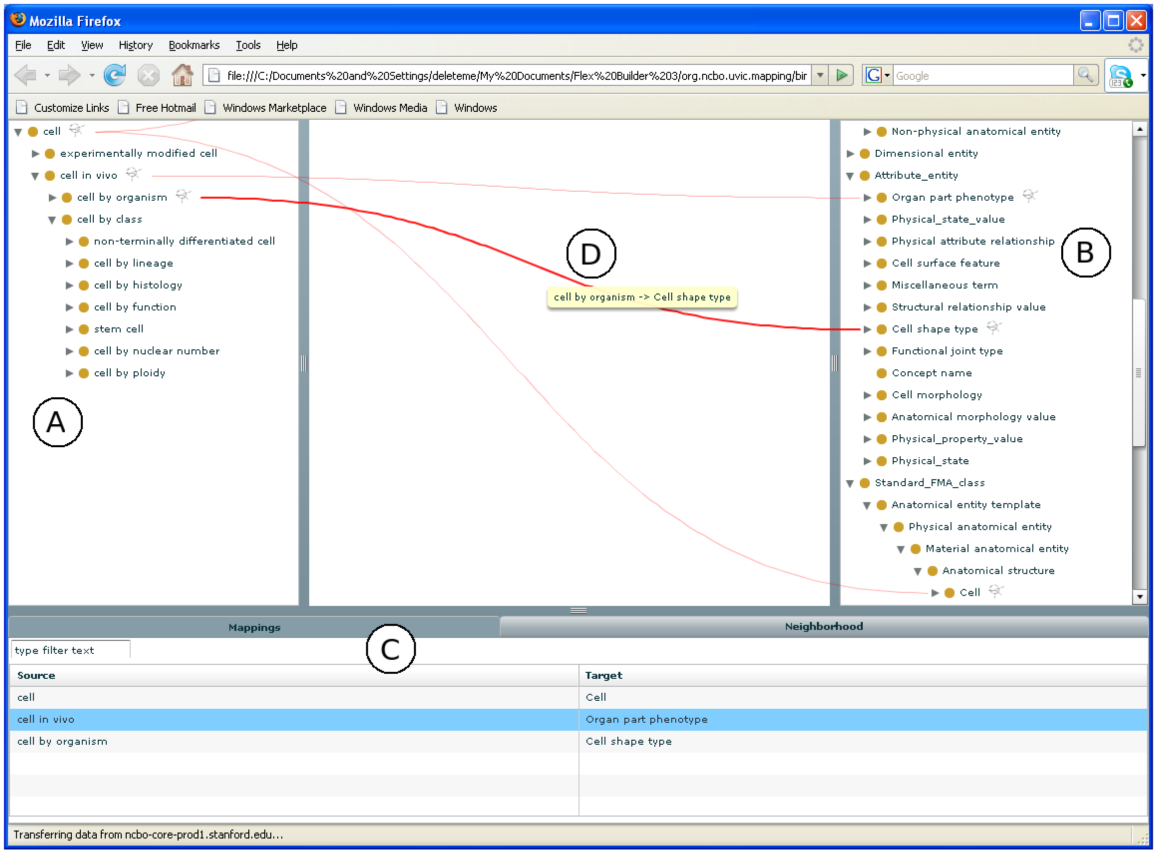

Figure 11.1. *Flex-based version of* COGZ. *(A) shows the source ontology, (B) the target ontology and (C) the list of correspondences in BioPortal. (D) shows a selected mapping line.*

actions. For example, expanding a concept in the source ontology results in a service call to BioPortal to retrieve that concept's immediate children. Once these are returned, the information is parsed and displayed in the source tree. This AJAX[2] approach allows the web-based COGZ to scale to very large ontologies as only parts of the entire ontology and correspondences are in memory and are based on the user's immediate requirements.

The current user interface is very similar to the Java-based COGZ interface (see Figure 11.1). Mappings are represented as arcs from source to target concepts. A list of all correspondences in the system are displayed at the bottom of the screen and are searchable. Also, a neighborhood view is available for structurally comparing concepts. This view inte-

[2] `http://en.wikipedia.org/wiki/Ajax_(programming)`

grates the graph-based visualizations that we previously developed for BioPortal. Like the desktop version of COGZ, selecting correspondences from the list are synchronized with the visualization and vice versa.

This system is still at a very early stage, but in the future, we would like to support mapping construction as well. The asynchronous nature of the application gives us a lot of potential to investigate exploration of very large ontologies like SNOMED CT.

11.2 Creating visualizations through ontology mapping

In our collaborations with the NCBO, we have been developing an information visualization toolkit. The goal of the visualization toolkit is to provide a general means of rapidly developing and deploying ontology-specific instance data visualizations to BioPortal. These visualizations are meant to assist scientists with exploring and understanding their data.

In previous work, Bull [Bul08] developed a model-based approach to generating visualizations called Model Driven Visualization (MDV). In this work, he represented visualizations (e.g., node-link diagrams, charts) with abstract software models using the Eclipse Modeling Framework (EMF) [BSM+03]. To use these visualizations, a software data model was "mapped" or transformed to the visual model, and then the corresponding visualization could be generated by executing the transformation.

We recognized that a similar approach could be used for generating visualizations of ontological instance data where the visualizations are constructed by mapping ontology concepts to view model concepts. In this work, we are exploring this idea by demonstrating how to leverage existing software modeling tools along with ontology mappings tools to generate highly customized visualizations. Before discussing the extensions made to COGZ to support this, we first must introduce relevant background from software modeling.

11.2.1 Model background

Model Driven Engineering (MDE) is an approach to software development by which software is specified, designed, implemented and deployed through a series of models and model transformations [Sel03, Sel06]. Software models, while designed and developed to assist engineers with the process of building quality software, share a number of commonalities with ontologies. Software models typically consist of classification hierarchies. For each concept in the hierarchy, its name, properties and relationship to other concepts can be specified. Software engineers use these models to help understand the domain, test hypotheses, build prototypes and even generate working systems. Toolkits have even emerged that allow knowledge and software engineers to convert their data between ontology and software model representations. We believe by integrating the concepts from both disciplines, we can leverage strengths of each discipline and enable the rapid specification and generation of customized visualizations of ontology instance data.

To build software using MDE, engineers first capture both domain concepts and their relationships as Platform Independent Models (PIMs). These models are transformed to one or more Platform Specific Models (PSMs), which can be executed. By developing software in this manner, the same PIM can be used on a number of different platforms. To support this software development methodology, a number of software modeling languages have emerged. Examples of such languages include the Unified Modeling Language (UML) [OMG07b] and the Eclipse Modeling Framework (EMF). In addition to software modeling languages, transformation languages have also been designed to facilitate the transformation from one model to another, or from PIM to PSM. In the software engineering world, the Atlas Transformation Language (ATL) [JK06] and Query View Transformation (QVT) [OMG07a] are widely used.

Bull investigated how information visualizations can be specified as a series of platform specific models [Bul08]. Using these models, he has successfully transformed complex software models to these visualization models, facilitating the rapid construction of highly customized visualizations. Because of the overlap between software modeling and ontol-

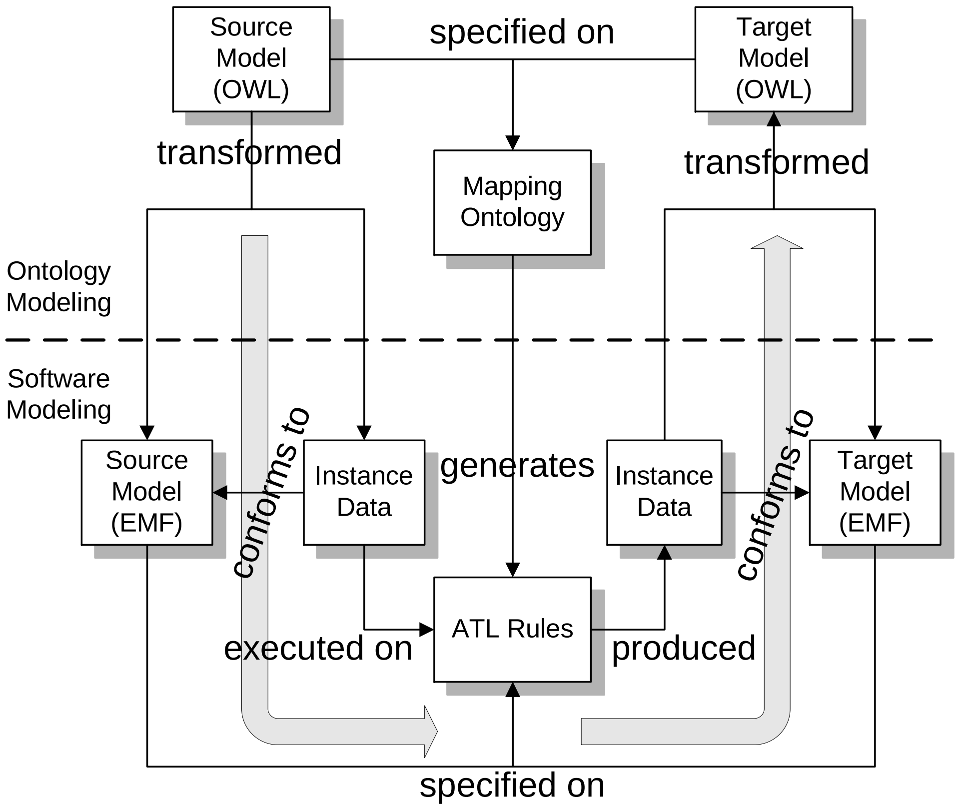

Figure 11.2. COGZ *instance data mediation architecture.*

ogy development, we are investigating if modeling technologies, and in particular Model Driven Visualization, can be applied to the process of customizing ontology visualizations.

11.2.2 Tool extensions

The extended COGZ architecture for ontology instance data mediation is shown in Figure 11.2. We use the existing EMF visualization models introduced by Bull [Bul08]. To make these usable within COGZ, we integrated the Eclipse Ontology Definition Metamodel (EODM) [3] project, which supports automatic conversion between EMF and OWL models. To support data integration between the source and target models, we developed an ATL rule writing library that converts the specified source to target correspondences that are stored in the mapping ontology by PROMPT into transformation rules that describe how to

[3] http://www.eclipse.org/modeling/mdt/?project=eodm#eodm

convert data between the two models. Since ATL only works with software models, we automatically convert the source and target OWL models into EMF and use the EMF versions for data transformation. The file conversions and generating process happen automatically and are hidden from the user.

The ATL rule writer library supports three types of correspondences: concept to concept, data type property to data type property, and reference property to reference property correspondences. We extended COGZ to support a property value editor for concept to concept correspondences. This allows the user to assign values to a target concept's property for a specific correspondence. This is necessary for manipulation of core visual elements. For example, assume we map a concept *User* from a source ontology to the concept *GraphNode* in a target visualization ontology. The *GraphNode* concept represents the node in a node-link graph visualization and we want all users in our ontology to be represented by a node. The *GraphNode* may have properties such as *color, size, shape, etc.* It is important that the end-user be able to customize the visualization to their needs, such as making the nodes in the graph that represent a user to be colored "red". The property editor allows the end-user to assign this value to the property and this will automatically be associated with the corresponding mapping rule, see Listing 11.1.

```
rule mappingRule {
 from
  IN : shrimpbib!User
 to
  OUT : distinct nodelink!GraphNode
      nodeLabel <- IN.nickname,
      color <- 'red'
}
```

Listing 11.1. *Example of User to GraphNode mapping rule*

Finally, we enhanced COGZ to allow the end-user to execute their transformation rules and generate the result as a visualization. By executing this, the instance data stored and associated with the source OWL model is converted into target model elements via the specified ATL rules. The Model Driven Visualization rendering engine is then used to dis-

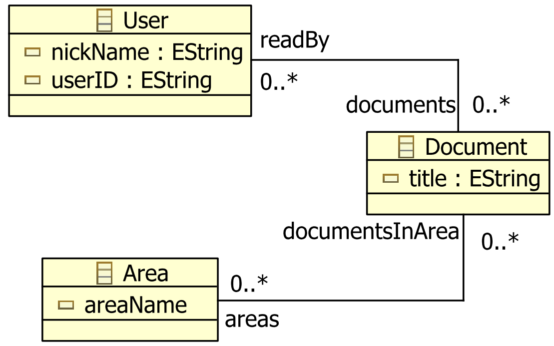

Figure 11.3. *A simplified representation of the domain ontology.*

play the visualization. The MDV rendering engine is capable of rendering any visualization conforming to one of the pre-defined models. Models exist for *node-link diagrams, charts, maps* and *treemaps*, among others. In the next section, we discuss a specific example demonstrating how this process works.

11.2.3 Case Study

To demonstrate the feasibility of this approach, we chose to regenerate an ontology instance data visualization discussed in Allen [All03] and Bull [Bul08]. Allen developed her own ontology that described researchers, research areas, research documents, and the relationships between these concepts (Figure 11.3). She programmed a specific visualization to generate a graph displaying researchers and the documents they have read. Bull demonstrated how to generate the same view using MDV. He converted Allen's ontology into an EMF model and then manually constructed ATL rules to transform the source model into his view model. In this case study, we demonstrate how to generate a similar visualization by simply dragging and dropping correspondences in CogZ between Allen's source ontology and Bull's visualization model.

The desired visualization is a graph-based view that connects researchers to their area of expertise based on the papers they have read. In the model, there is no direct association between a *User* and an *Area* (this information can only be derived by looking at the papers

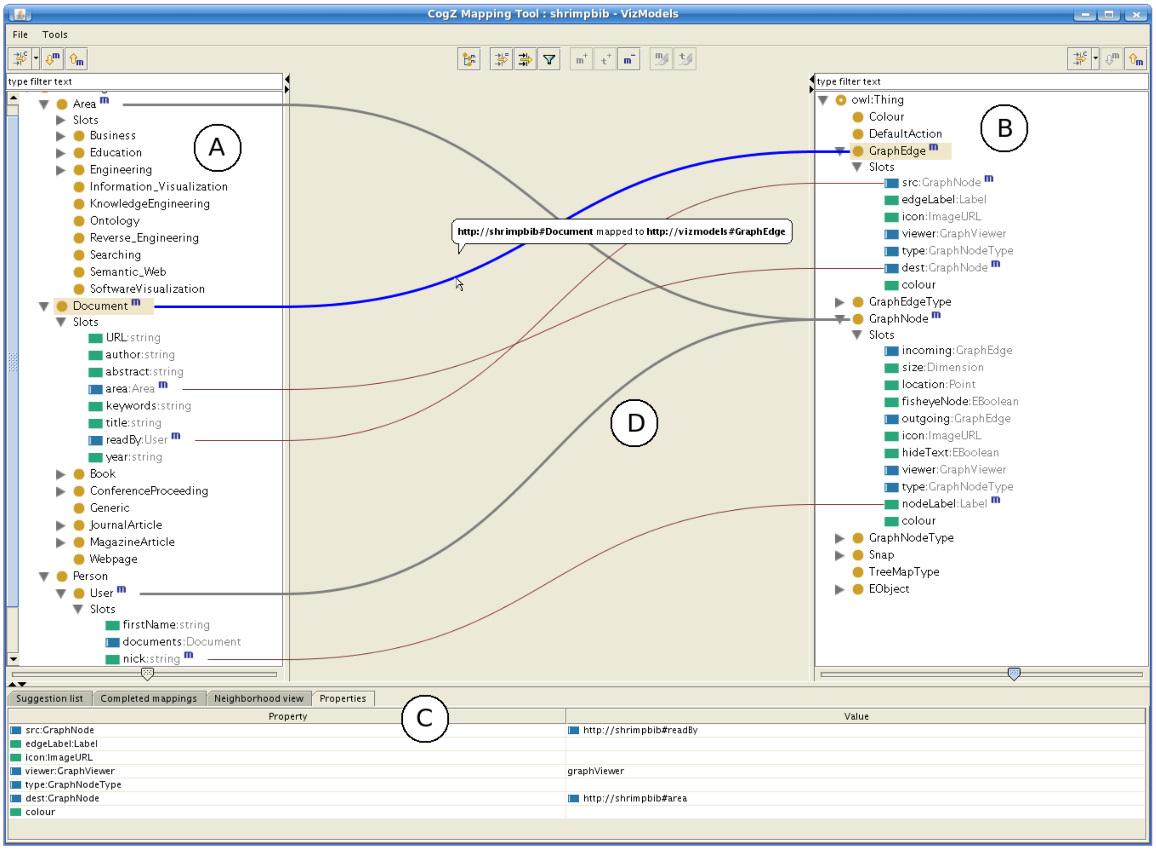

Figure 11.4. *Mappings from domain ontology to visualization ontology. (A) shows the source domain ontology, (B) shows the target view ontology, (C) shows the property editor for a mapping, and (D) shows the visual representation of correspondences. Thick arcs represent concept to concept correspondences and thin arcs represent property correspondences.*

each researcher has read). In the following, we demonstrate how a mapping can be used to describe the desired visualization.

To begin, we load the COGZ tool and specify Allen's model as our source ontology and Bull's node-link model as our target. We know that we want to represent both users and areas as nodes and link the data elements based on representing documents as edges. To do this, we drag mapping lines from the source concept *User* to the target concept *GraphNode*, *Area* to *GraphNode*, and finally *Document* to *GraphEdge*. We also assign several data type property correspondences to configure the labels for the *GraphNode* correspondences. We map the *User* property *nickname* to *GraphNode*'s *nodeLabel* property and *Area*'s *name*

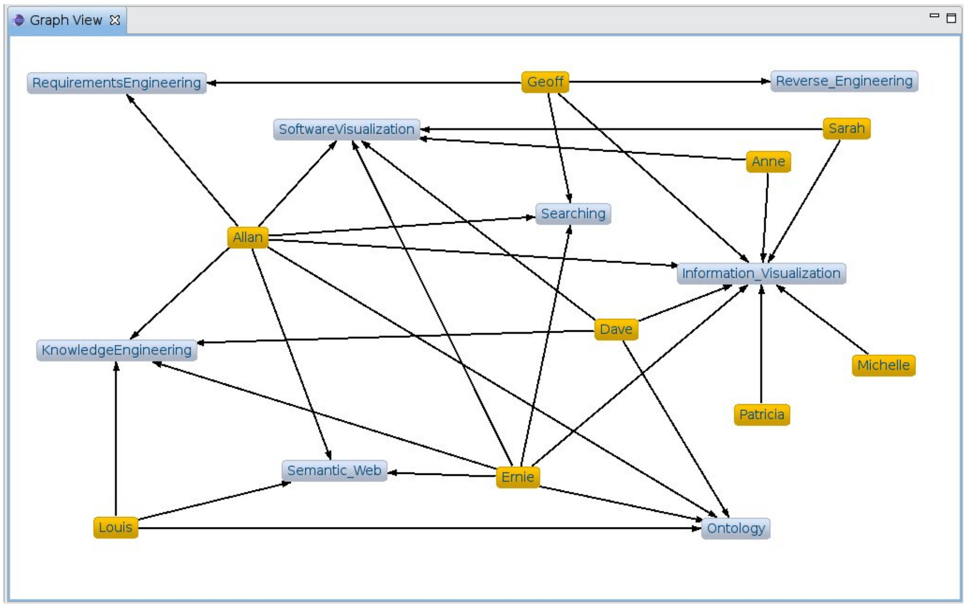

Figure 11.5. *Visualization generated by mapping rules. Researchers are connected to their areas of expertise based on documents they have read from those areas.*

property to *nodeLabel*. An ATL rule is automatically created for each concept to concept correspondence, similar to Listing 11.1. The rule describes which source class is being mapped to a target class and which properties should be mapped between the two concepts. Each of these correspondences are simple one-to-one mapping correspondencess and the rule generation is straight-forward.

The final correspondences necessary to construct the view are the reference mappings from a *Document*'s reference to the areas it is contained in and the reference to the user's that have read that particular document. We map the *area* reference property to the *GraphEdge*'s *destination* property, while the *readyBy* property is mapped to the *GraphEdge*'s *source* property. The complete mapping as represented by COGZ is displayed in Figure 11.4.

The reference property mappings have different cardinalities as both the *area* and *readBy* properties can have multiple values while the *destination* and *source* represent a single instance. This change in cardinality is automatically detected by our ATL rule writing engine and resolved by computing a cross-product between the users mapped to *source* and areas mapped to *destination*. An example of the rule generated for the *Document* to *GraphEdge* mapping is shown in Listing 11.2.

The mappings necessary to generate the view are now specified and the transformation rules can be executed. The resulting visualization is shown in Figure 11.5.

```
rule mappingRule {
 from
  IN : shrimpbib!Document
 to
  OUT : distinct nodelink!GraphEdge
    foreach( e in IN.computeCrossProduct() ) (
        edgeLabel <- IN.title ,
        dest <- e.domain,
        src <- e.range
    )
}
```

Listing 11.2. *Document to GraphEdge mapping rule*

11.2.4 Discussion

We extended the COGZ ontology mapping tool to support the generation and customization of ontology instance data visualizations. These extensions involved combining techniques from both software modeling and ontology modeling. We extended work from Model Driven Visualization by incorporating existing EMF visualization models as our target OWL view models. We integrated COGZ with the MDV toolkit for executing ATL rules and generating visualizations. We demonstrated the feasibility of this approach through a case study where we recreated a previously developed ontology instance data visualization simply by specifying correspondences from a source to target ontology.

We believe this approach is very promising for the specification and generation of visualizations. This approach gives us the generality we require for the visualization toolkit we are developing for BioPortal. Potentially, correspondences that specify a visualization could be shared in BioPortal and end-users could modify the mappings and property values to create their own custom visualizations.

Furthermore, these extensions help to "inform" our cognitive support framework for use in a specific application of mapping. Specifically, the case study and process of extending the tool has helped us evaluate the flexibility of our approach outside of the application of mapping that we originally designed the tool for. In the future, there is still a lot of work that could be done. The problem of supporting the process of defining transformation rules for mapping is a topic that deserves an entire thesis in itself. We believe our experimental approach, framework, and tool could be useful with this research.

11.3 Summary

In this chapter, we briefly introduced some of the extensions to COGZ that have followed from the design elements introduced by our framework. We believe the web-based COGZ project has potential outside of BioPortal with assisting users with exploring large ontologies and mappings online. The custom tools developed by the mapping team discussed in Chapter 7 were web-based, but they have to explore the mappings off-line in spreadsheets. The Flex version of COGZ could potentially be adapted to integrate with their tools to help them explore and validate the mappings they generate. The generation of visualizations through mappings project that was discussed helped demonstrate the flexibility of the COGZ tool to different applications of mapping. Moreover, besides showing an exciting way to construct and develop new visualizations for ontologies, this approach could also potentially be used to support users with performing general software model mapping.

Chapter 12

Conclusions

In this chapter we first present a number of future directions for this research. We discuss improvements to the evaluation, expand on collaborative aspects of mapping, and improvements to our behaviorial studies. We also look at applying this research beyond ontology mapping and into other domains of mapping. Following this we present and summarize the contributions of this thesis. We believe that one of our primary contributions is helping to establish and draw attention to the problem of incorporating a user into the mapping process.

12.1 Future work

While we have performed a number of experiments investigating the user's role in the ontology mapping process, there is still a lot work that needs to be done in this area. There is also much more that can be developed in terms of tool support. Below we describe a number of directions we believe this work could follow.

12.1.1 Tool evaluation

We believe the tool evaluation experiment discussed in Chapter 10 is a big step forward for this domain in terms of user support evaluation. However, there were limitations to our experiment that could be removed with future studies. For example, our study only included two expert users. In the future, it would be interesting to carry out a case study series with real ontology mapping users where they use CogZ for a longer duration of time. This would hopefully uncover other types of mapping issues, such as constructing a mapping over multiple sessions and team coordination. It may also be interesting to continue the original study with more users, which would make it easier to apply other types of quantitative analysis approaches, such as uncovering specific differences on certain tasks between the tools and comparing across user populations.

12.1.2 Team mapping

Both our survey and interview study demonstrate a need for team-based ontology mapping tools. This would be an interesting direction to explore in the future, perhaps building on some of the existing work in social computing. Like software development, ontology mapping is largely a team-based process. Future research needs to explore how collaborative correspondences can be constructed and what elements of team awareness need to be supported.

12.1.3 Behavioral studies

There are many interesting directions that the studies discussed in Chapter 3 on human inference could go in the future. For example, it could be useful to analyze the divergence between automated mapping procedures and human mapping to try to understand the fundamental differences between how machines and humans compare information. This analysis potentially would yield insights into human inference and judgement along with potential ideas for improving existing mapping algorithms.

12.1.4 Beyond ontology mapping

In Chapter 11, we discussed extensions that we made to CogZ to support software model mapping. This extension builds on previous work from software modeling. We believe the transformation approach could potentially be used for general model mapping. However, this is really only a starting point for supporting model mapping. In the future, it would be interesting to explore how to develop mapping algorithms specific for data integration. PROMPT's default algorithm does not perform well for this type of mapping scenario. PROMPT relies primarily on lexical similarities between the models. However, with this particular application, it is unlikely that the source model will have concept names lexicographically similar to the view model. We believe a more appropriate algorithm would put a higher priority on data type comparison once some concept to concept correspondences are manually generated.

Bernstein *et al.* [BMC06] propose a similar incremental approach for schema matching. To help reduce the complexity of validating a large number of potential correspondences from an entire schema, the user directs the mapping process by computing correspondences one term at a time. The algorithm uses this information to infer other suggestions. Although PROMPT follows a similar incremental approach, it is not completely directed by the user. It would be interesting to explore how COGZ's filtering and visualization support could be combined with this purely incremental approach.

It would also be interesting to investigate the database mapping community. Although this thesis was focused on a specific type of ontology mapping, we believe there is overlap between these two fields. For example, similar to ontologies, database schemas can be represented as graphs where the edges are relationships between tables and table fields and the nodes are the entities of the schema. Typically, database schemas have less structure than ontologies. However, there are relationships between tables represented explicitly via foreign key constraints or implicitly through the data.

Database schemas also often have instance data associated with them. This data can be a huge help with automatically generating correspondences as the data can be analyzed and compared to help provide more information about the validity of a candidate correspondence. From a use perspective, the data is also a powerful tool to help validate a potential correspondence. If by inspection the data is highly similar, then this is a strong indicator that the terms should be mapped.

Like ontology mapping, schemas can be large and difficult to understand and different databases storing similar information may have been designed with different levels of granularity. As a result, many of the design principles for ontology mapping tools that we introduced may also be applicable in the database mapping community.

12.2 Contributions

Several contributions have emerged from the work presented in this thesis. These contributions are discussed below and divided into two categories: scientific and engineering contributions.

12.2.1 Scientific contributions

12.2.1.1 Exploratory studies

To our knowledge, there have been no user studies carried out that analyze how users make decisions with mapping tools, what their processes are, and how they work in teams. The studies discussed in this thesis are a starting point for other researchers interested in conducting similar research. Moreover, the human factors outlined in Chapter 3 can help guide other studies in this community.

12.2.1.2 Drawing attention to the human in the loop

The results and publications from the studies discussed in this thesis help draw attention to the important role that the user performs during the ontology mapping process. We propose that aiding and supporting the user with their decisions is key to improving the quality and adoption of mappings. Drawing attention to this important issue is our primary contribution to the mapping community.

12.2.1.3 Cognitive support framework

The cognitive support framework helps to summarize the results from the various exploratory and behavioral studies. It can be used to relate separate research communities: cognitive psychology, human computer interaction, information visualization, and the semantic web. The framework provides a universal description of the concepts involved in the process of ontology mapping and a set of design principles for developing mapping tools. Other researchers can use this framework to help guide the design of their own tools.

12.2.2 Engineering contributions

12.2.2.1 Plugin framework

The plugin framework we developed for PROMPT benefits the ontology mapping community as it provides a way for researchers and developers to target specific parts of the mapping process, but still have the full support of a comprehensive mapping tool. Also, the alternative perspectives supported by the framework aids researchers with comparing different user interfaces for mapping while keeping the back-end systems identical.

12.2.2.2 COGZ tool

COGZ is an implementation of the design principles introduced by the cognitive support framework. As mentioned in Chapter 10, the tool has been adopted by a number of researchers, for example both Gennari *et al.* [GNCC08] and Gil *et al.* [GBRC08] have used COGZ's neighborhood view for visual validation of correspondences. Although similar line-based mapping interfaces exist, our work has shown that it can be difficult to get the design right. Previous work by Robertson *et al.* [RCC05b] discuss the difficulty with developing this type of display. One key improvement in our approach is the synchronization of the line drawings with user operated filters. Without this feature the user interface does not scale and quickly overloads the user.

12.2.2.3 Tool evaluation

The controlled study discussed in Chapter 10 is the first study in this domain that attempts to evaluate mapping tools in terms of the cognitive support provided by the tool. The findings from the study contribute to a theory of required tool support. Previous studies that have explored the human aspects of data alignment are not only sparse, but also only measure effectiveness without an explanation of the results. This is not surprising as no theory on how users define a mapping existed at that time. Our study is a step in forming a theory on the kind of cognitive support user centric mapping requires. This study also contributed to the field of Human Computer Interaction (HCI), as there are few works in this area that map theoretical concepts from HCI to real artefacts.

12.3 Summary

The semantic web brings structure and formal semantics to web data. The vision is to create a globally linked database of information, where data can be shared between web pages and local data stores [Pal01]. A prerequisite for information sharing is the mapping of independent data representations. This procedure is usually carried out offline and relies on the knowledge of domain experts. Throughout this thesis, we have advocated for cognitive support in ontology mapping tools. Existing research points to a tendency to think of the underlying ontology mapping algorithm as mostly independent from the user. We strongly believe that by embracing a unified view of human and machine, cognitive aids introduced to the mapping process will enhance the quality of mappings.

In the ontology research community, there have been limited usability experiments and user-based evaluations. Also, within the human computer interaction community, the general problem of structured terminology integration has been largely ignored despite its importance for data management. We need to understand how interfaces for mapping can be improved and how to evaluate these approaches.

We have attempted to address these issues by carrying out and examining a number of exploratory and behavioral user studies that investigated how human inference influences our judgements about mapping correspondences, how users interact with mapping tools, what problems arise, which tools are widely used, what process is followed during mapping, and how teams carry out a mapping. We used the results from these various experiments to propose a cognitive support framework that described the information needs of ontology mapping users, the process they follow, and proposed a number of design requirements for mapping tools.

Based on these requirements we iteratively developed a tool called COGZ. The tool combines visualization techniques with a variety of filtering mechanisms to help support users explore and validate a mapping. We evaluated the tool by first demonstrating the feasibility of the approach through a case study where we used an existing anatomy mapping between two large widely used ontologies. Following this, we carried out a controlled lab experiment comparing the COGZ perspective to the PROMPT perspective on a number of tasks. We explored why COGZ was preferred and why users performed better with it in terms of the cognitive support provided by the tool. We believe the design of this study can be used as a template for other researchers interested in evaluating the user support their tool provides for ontology mapping. Moreover, the new alternative perspective feature we introduced into PROMPT gives researchers a unique opportunity to use an identical back-end process during the experiment so that the focus is on the user interface. Finally, we discussed the adoption of the tool and framework that has taken place throughout this research.

The overall theme is that tools encode a workflow process and that this process must align with the user's own internal process. By aligning these processes, we will be able to assist rather than hinder the user. We must incorporate a "human in the loop", where the human is an essential component in the mapping process. Helping to establish and harness this symbiotic relationship between human processes and the tool's automated process will allow people to work more efficiently and effectively, and afford them the time to concentrate on difficult tasks that are not easily automated.

References

[ABC98] David Abrams, Ronald Baecker, and Mark H. Chignell. Information archiving with bookmarks: Personal web space construction and organization. In *Human Factors in Computing Systems (CHI 98)*, pages 41–48. ACM Press, 1998.

[All03] Mary Margaret Allen. Empirical evaluation of a visualization tool for knowledge engineering. Master's thesis, University of Victoria, 2003.

[Bai21] Nathaniel Bailey. *An Universal Etymological English Dictionary*. 1721.

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[BG00] Dan Brickley and R. V. Guha. Resource description framework (rdf) schema sepcficiation 1.0. http://www.w3.org/TR/2000/CR-rdf-schema-20000327/, 2000.

[Bil91] Charles E. Billings. Human-centered aircraft automation: A concept and guidelines. Technical Report 103885, Ames Research Center, 1991.

[biz07] Understanding biztalk messaging services. http://www.perfectxml.com/syngress/biztalk/Page1.asp, April 2007.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific America*, May 2001.

[BM07] Philip A. Bernstein and Sergey Melnik. Model management 2.0: Manipulating richer mappings. In *SIGMOD'07*, pages 1–12, June 2007.

[BMC06] Philip A. Bernstein, Sergey Melnik, and John E. Churchill. Incremental schema matching. In *VLDB '06: Proceedings of the 32nd international Conference on Very Large Databases*, pages 1167–1170. VLDB Endowment, 2006.

[Bor96] Stephen P. Borgatti. Elements of a theoretical framework. http://www.analytictech.com/mb313/elements.htm, 1996.

[Bot89] M. Elaine Botha. Theory development in perspective: the role of conceptual frameworks and models in theory development. *Journal of Advanced Nursing*, 14(1):49–55, 1989.

[BP] Bioportal. http://bioportal.bioontology.org/.

[Bra83] Lisanne Brainbridge. Ironies of automation. *Automatica*, 19:775–779, 1983.

[Bro57] Roger Brown. Linguistic determinism and the part of speech. *Journal of Abnormal and Social Psychology*, 55:1–5, 1957.

[Bro96] James Brooke. Usability evaluation in industry. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, editors, *SUS: a 'quick and dirty' usability scale,*, pages 184–194. Taylor & Francis, 1996.

[BSM+03] Frank Budinsky, David Steinberg, Ed Merks, Raymond Ellersick, and Timothy J. Grose. *Eclipse Modeling Framework*. Addison Wesley, 2003.

[Bul08] Robert Ian Bull. *Model Driven Visualization: Towards a Model Driven Engineering Approach for Information Visualizations*. PhD thesis, University of Victoria, 2008.

[CFF+98] V. K. Chaudhri, A. Farquhar, R. Fikes, P. D. Kar, and J. P. Rice. Open knowledge base connectivity 2.0.3, specification document. Technical report, 1998.

[CM03] M. Crubézy and M. A. Musen. Ontologies in support of problem solving. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 321–342. Sringer, 2003.

[Cog] Cognitive load. http://en.wikipedia.org/wiki/Cognitive_load.

[Cor08] Raul Corazzon. Ontology. a resource guide for philosophers. http://www.formalontology.it/, 2008.

[Cre03] John W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2 edition, 2003.

[cry] Crystal reports. http://www.businessobjects.com/products/reporting/crystalreports.

[CS06] Silvia Coradeschi and Alessandro Saffiotti. Symbiotic robotic systems: Humans, robots, and smart environments. *IEEE Intelligent Systems*, 21(3):82–84, 2006.

[D02́] Françoise Détienne. *Software Design - Cognitive Aspects*. Springer-Verlang, New York, NY, USA, 2002.

[DdB+08] Chan Le Duc, Mathieu dAquin, Jesus Barrasa, Jérôme David, Jérôme Euzénat, Raul Palma, Rosario Plaza, Marta Sabou, and Boris Villazn-Terrazas. Matching ontologies for context: The neon alignment plug-in. Technical Report Deliverable 3.3.2, IST NeOn IP, NeOn, 2008.

[DDH03] AnHai Doan, Pedro Domingos, and Alon Halevy. Learning to match the schemas of data sources: A multistrategy approach. *Machine Learning*, 50(3):279–301, March 2003.

[DF02] Ying Ding and Schubert Foo. Ontology research and development. part 2 - a review of ontology mapping and evolving. *Journal of Information Science*, 28(5):375–388, 2002.

[Do06] Hong-Hai Do. *Schema Matching and Mapping-based Data Integration*. PhD thesis, Department of Computer Science, Universität Leipzig, 2006.

[dP04] Jos de Bruijn and Axel Polleres. Towards an ontology mapping specification language for the semantic web. Technical report, Digital Enterprise Research Institute, 2004.

[Ee04] Jérôme Euzénat and et al. State of the art on ontology. deliverable d2.2. Technical report, Knowledge web NoE, 2004.

[EH81] H. J. Einhorn and R. M. Hogarth. Behavioral decision theory: Processes of judgement and choice. *Annual Review of Psychology*, 32:53–88, 1981.

[ELTV04] Jérôme Euzénat, David Loup, Mohamed Touzani, and Petko Valtchev. Ontology alignment with ola. In *Proceedings of the 3rd EON Workshop, 3rd International Semantic Web Conference*, November 2004.

[End95] Mica R. Endsley. The out-of-the-loop performance problem and level of control in automation. *Human Factors*, 37(2):381–394, 1995.

[ES80] K. Ericsson and H. Simon. *Psychological Review*, 87(3):215–251, 1980.

[ES04a] M. Ehrig and S. Staab. Qom - quick ontology mapping. In *Proc. of the Third International Semantic Web Conference (ISWC2004)*, 2004.

[ES04b] M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In *1st European Semantic Web Symposium*, Heraklion, Greece, 2004. Springer, LNCS.

[ES05] Marc Ehrig and York Sure. FOAM–framework for ontology alignment and mapping; results of the ontology alignment initiative. In *Proceedings of the Workshop on Integrating Ontologies*, volume 156, pages 72–76, October 2005.

[ESA05] Neil A. Ernst, Margaret-Anne Storey, and Polly Allen. Cognitive support for ontology modeling. *International Journal of Human-Computer Studies*, 62(5):553–577, 2005.

[etl] Etl tool survey. http://www.etltool.com/index.htm.

[Euz06] Jérôme Euzénat. *An API for ontology alignment (version 2.1)*, February 2006.

[FBGS09] Sean M. Falconer, R. Ian Bull, Lars Grammel, and Margaret-Anne Storey. Creating visualizations through ontology mapping. In *2nd International Workshop on Ontology Alignment and Visualization*, 2009. Accepted.

[FBS] Sean M. Falconer, Gargi Bougie, and Margaret-Anne Storey. A case study of an ontology mapping team. Under review, OnAV'09.

[FNS06] Sean M. Falconer, Natalya F. Noy, and Margaret-Anne Storey. Towards understanding the needs of cognitive support for ontology mapping. In *Ontology Matching Workshop*, 2006.

[FNS07] Sean M. Falconer, Natalya F. Noy, and Margaret-Anne Storey. Ontology mapping - a user survey. In *Ontology Matching Workshop*, November 2007.

[FP02] Jean-Daniel Fekete and Catherine Plaisant. Interactive information visualization of a million items. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 117, Washington, DC, USA, 2002. IEEE Computer Society.

[FS07a] Sean M. Falconer and Margaret-Anne Storey. Cognitive support for human-guided mapping systems. Technical Report DCS-318-IR, University of Victoria, Victoria, BC, Canada, 2007.

[FS07b] Sean M. Falconer and Margaret-Anne Storey. A cognitive support framework for ontology mapping. In *Proceedings of ISWC/ASWC*, pages 114–127, November 2007.

[FYS] Sean M. Falconer, Takashi Yamauchi, and Margaret-Anne Storey. Cogz: Evaluating cognitive support for semi-automatic terminology mapping. Under review, CHI'09.

[GBRC08] Richard Gil, Ana María Borges, Luís Ramos, and Leonardo Contreras. Ontologies integration for university institutions: Approach to an alignment evaluation. In *ASWEC '08: Proceedings of the 19th Australian Conference on Software Engineering (aswec 2008)*, pages 570–578, Washington, DC, USA, 2008. IEEE Computer Society.

[GF92] Michael R. Genesereth and Richard E. Fikes. Knowledge interchange format: Version 3.0 reference manual, 1992.

[GH99] Susan A. Gelman and Gail D. Heyman. Carrot-eaters and creature-believers: The effects of lexicalization on children's inferences about social categories. *Psychological Science*, 10:489–493, 1999.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. W.H. Freeman Company, 1979.

[GM86] Susan A. Gelman and Ellen M. Markman. Categories and induction in young children. *Cognition*, 23:183–209, 1986.

[GNCC08] John H. Gennari, Maxwell L. Neal, Brian E. Carlson, and Daniel L. Cook. Integration of multi-scale biosimulation models via light-weight semantics. In *Pacific Symposium on Biocomputing*, pages 414–425, 2008.

[GPP+93] Nicola Guarino, Roberto Poli, Kluwer Academic Publishers, In Press Substantial, and Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In *International Workshop on Formal Ontology*, March 1993.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, 1993.

[Gru09] Tom R. Gruber. Ontology. In *Encyclopedia of Database Systems*. Springer-Verlag, 2009. To appear.

[Ham79] James A. Hampton. Polymorphous concepts in semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 18:441–481, 1979.

[HB00] Evan Heit and Lewis Bolt. Knowledge selection in category learning. *The Psychology of Learning and Motivation*, 39:163–199, 2000.

[HIB] hibernate.org. http://www.hibernate.org/.

[HMK08] David Huynh, Robert Miller, and David Karger. volume 4825, pages 239–252. Springer-Verlag, 2008.

[Hor00] Ian Horrocks. A denotational semantics for oil-lite and standard oil. Technical report, Department of Computer Science, University of Manchester, UK, 2000.

[Hut95] E. L. Hutchins. *Cognition in the wild*. MIT Press, 1995.

[Hv01] Ian Horrocks and Frank van Hamelen. Reference description of the daml+oil ontology markup language. Technical report, 2001.

[Ins03] Electronics & Telecommunications Research Institute. Moa shell. http://mknows.etri.re.kr/moa/docs/moashell.html, 2003.

[JK06] Frédéric Jouault and Ivan Kurtev. Transforming Models with ATL. *Lecture Notes in Computer Science, Springer Berlin / Heidelberg*, 3844:128–138, 2006.

[Jon72] Karen Sprck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[JZZ] Jazz community site. https://jazz.net/pub/index.jsp.

[KCT] Peter D. Karp, Vinay K. Chaudhri, and Jerome Thomere. Xol: An xml-based ontology exchange language. Technical report, Pangea Systems Inc. and SRI International.

[KD62] A. C. Kerckhoff and K. E. Davis. Value consensus and need complementarity in mate selection. *American Sociological Review*, 27:295–303, 1962.

[KD08] Ravikanth Kolli and Prashant Doshi. Optima: Tool for ontology alignment with application to semantic reconciliation of sensor metadata for publication in sensormap. *ICSC*, pages 484–485, 2008.

[Kei] Frank C. Keil. Folkscience: coarse interpretations of a complex reality. *Trends in Cognitive Sciences*, 7:368–373.

[Kuh55] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quaterly*, 2:83–95, 1955.

[Lab] The Jackon Laboratory. Adult mouse anatomical dictionary browser. http://www.informatics.jax.org/searches/AMA_form.shtml.

[LE03] Patrick Lambrix and Anna Edberg. Evaluation of ontology merging tools in bioinformatics. In *Proceedings Pacific Symposium on Biocomputing*, pages 589–600, Kauai, Hawaii, USA, 2003.

[Len95] Douglas B. Lenat. Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):33–38, 1995.

[LG89] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[Lor06] Jacob Lorhard. *Ogdoas Scholastica*. Sangalli : Apud Georgium Straub, 1606.

[Löw93] J. Löwgren. *Human-computer Interaction. What every system developer should know.* Studentlitteratur, Lund, 1993.

[LS06] Monika Lanzenberger and Jennifer Sampson. Alviz - a tool for visual ontology alignment. In *IV '06: Proceedings of the conference on Information Visualization*, pages 430–440, Washington, DC, USA, 2006. IEEE Computer Society.

[Mar89] Ellen M. Markman. *Categorization and naming in children*. MIT Press, 1989.

[McC80] John McCarthy. Circumscription – a form of non-monotonic reasoning. *Artificial Intelligence*, 5(13), 1980.

[Mel04] Sergey Melnik. *Generic Model Management: Concepts and Algorithms*. PhD thesis, University of Leipzig, 2004.

[MFRW00] Deborah L. McGuinness, Richard Fikes, James Rice, and Steve Wilder. The chimaera ontology environment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 1123–1124. AAAI Press / The MIT Press, 2000.

[MHH00] Renée J. Miller, Laura M. Haas, and Mauricio A. Hernández. Schema mapping as query discovery. In *VLDB 2000*, pages 77–88, 2000.

[Mil] George Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63:81–97.

[MST08] C. Meilicke, H. Stuckenschmidt, and A. Tamilin. Supporting manual mapping revision using logical reasoning. In *Proceedings of the 23rd Conference on Artificial Intelligence (AAAI-08)*, 2008.

[Mur02] Gregory L. Murphy. *The Big Book of Concepts*. The MIT Press, 2002.

[NCB] The national center for biomedical ontology. http://www.bioontology.org/.

[nci08] Termintology resources - national cancer institute. http://www.nci.nih.gov/cancertopics/terminologyresources, 2008.

[NE008] Neon wiki, 2008.

[NFF$^+$91] Robert Neches, Richard Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3), 1991.

[NGM08] Natalya F. Noy, Nicholas Griffith, and Mark A. Musen. Collecting community-based mappings in an ontology repository. Technical Report BMIR-2008-1310, Stanford Center For Biomedical Informatics Research, 2008.

[NLP] An introduction to nlp. http://www.cs.bham.ac.uk/ pxc/nlpa/2002/AI-HO-IntroNLP.html.

[NM02] Natalya F. Noy and Mark A. Musen. Evaluating ontology-mapping tools: Requirements and experience. In *In Proceedings of OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management*, pages 1–14, 2002.

[NM03] N. F. Noy and M. A. Musen. The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

[Nor93] D. A. Norman. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison-Wesley, 1993.

[Nor94] Donald A. Norman. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison Wesley Publishing Company, 1994.

[NS72] A. Newell and Herbert A. Simon. *Human problem solving*. Prentice Hall, Englewood Cliffs, NJ, 1972.

[NS07] Letitia R. Naigles and Lauren D. Swensen. *Syntactic Supports for Word Learning*, pages 212–231. Blackwell Handbook of Language Development. Blackwell Publishing Ltd, 2007.

[OAE06] Ontology alignment evaluation initiative. http://oaei.ontologymatching.org/2006/, 2006.

[ØAS05] Peter Øhrstrøm, Jan Andersenn, and Henrik Schärfe. What has happened to ontology, 2005.

[OBO] The open biomedical ontologies. http://www.obofoundry.org/.

[OM206] International workshop on ontology matching. http://om2006.ontologymatching.org/, 2006.

[OMG07a] OMG/QVT. MOF – Query / View / Transformation (QVT), 2007. available at http://www.omg.org.

[OMG07b] OMG/UML. Unified Modeling Language (UML), version 2.1.1, (Formal/2007-02-03), 2007. available at http://www.omg.org.

[ØSU05] Peter Øhrstrøm, Henrik Schärfe, and Sara L. Uckelman. Jacob lorhards ontology: A 17th century hypertext on the reality and temporality of the world of intelligibles, 2005.

[OWLa] Owl web ontology language overview. http://www.w3.org/TR/owl-features/.

[OWLb] Web ontology language (owl). http://www.w3.org/2004/OWL/.

[Pal01] Sean B. Palmer. The semantic web: An introduction. http://infomesh.net/2001/swintro/, September 2001.

[PB85] Frederick P. Brooks, Jr. *The Mythical Man-Month*. 1985.

[PC86] R. E. Petty and J. T. Cacioppo. *Communication and persuasion: Central and peripheral routes to attitude change*. Springer-Verlang, New York, NY, USA, 1986.

[PPP] Performance, parsing and pragmatics. http://www.phon.ucl.ac.uk/home/marco/Principles2006-2007.htm.

[PRS+94] Jenny Preece, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland, and Tom Carey. *Human-Computer Interaction*. Addison-Wesley, 1994.

[PT2] Protege ontology library. http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library.

[RCC05a] George G. Robertson, Mary P. Czerwinski, and John E. Churchill. Visualization of mappings between schemas. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 431–439, New York, NY, USA, 2005. ACM Press.

[RCC05b] George G. Robertson, Mary P. Czerwinski, and John E. Churchill. Visualization of mappings between schemas. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 431–439, New York, NY, USA, 2005. ACM.

[RL02] Stephen L. Reed and Douglas B. Lenat. Mapping ontologies into cyc. In *Proceedings of AAAI*, 2002.

[Ros78] Eleanor Rosch. *Principles of Categorization*, pages 27–48. John Wiley & Sons Inc, 1978.

[RR96] Jason Elliot Robbins and David F. Redmiles. Software architecture design from the perspective of human cognitive needs. In *Proceedings of the California Software Symposium (CSS'96)*, pages 16–27, 1996.

[SDH+86] Herbert A. Simon, George B. Dantzig, Robin Hogarth, Charles R. Piott, Howard Raiffa, Thomas C. Schelling, Kennth A. Shepsle, Richard Thaier, Amos Tversky, and Sidney Winter. Decision making and problem solving. *National Academy of Sciences*, 1986.

[SE05] Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics IV*, pages 146–171, 2005.

[SE08] Pavel Shvaiko and Jérôme Euzénat. Ten challenges for ontology matching. In *ODBASE*, 2008.

[Sel03] Bran Selic. The Pragmatics of Model-Driven Development. *IEEE Software*, 20(5):19–25, 2003.

[Sel06] Bran Selic. Model-Driven Development: Its Essence and Opportunities. In *Proceedings of the Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 313–319, 2006.

[SG2] Swoogle: semantic web search. http://swoogle.umbc.edu/.

[Shn92] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[SM97] João Setubal and João Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Company, 1997.

[SMD] Snomed clinical terms(snomed ct). http://www.nlm.nih.gov/snomed.

[SMD08] *SNOMED Clinical Terms User Guide*, 2008.

[SNM+02] Margaret-Anne Storey, Natasha F. Noy, Mark Musen, Casey Best, Ray Fergerson, and Neil Ernst. Jambalaya: an interactive environment for exploring ontologies. In *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces*, pages 239–239, New York, NY, USA, 2002.

[SW] W3c semantic web activity. http://www.w3.org/2001/sw/.

[Swe88] John Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(1):257–285, 1988.

[Swe94] John Sweller. Cognitive load theory, learning difficulty, and instructional design, learning and instruction. *Learning and Instruction*, 4(4):295–312, 1994.

[Syq] Syque. Theories about decision-making. http://changingminds.org/explanations/theories/a_decision.htm.

[Tha89] P. Thagard. Explanatory coherence. *Behavioral and Brain Sciences*, 12:435–467, 1989.

[TS04] Thomas S. Tullis and Jacqueline N. Stetson. A comparison of questionnaires for assessing website usability. In *Usability Pressionals Association (UPA) 2004 Conference*, 2004.

[Tun07] Nwe Ni Tun. *Semantic Enrichment in Ontologies for Anaysis and Matching*. PhD thesis, Japan Advanced Institute of Science and Technology, March 2007.

[TW] H. Tajfel and A. L. Wilkes. Classification and quantitative judgment. *British Journal of Psychology*, 54:101–114.

[UML] Unified medical language system (umls). http://www.nlm.nih.gov/research/umls/.

[Vic05] Bret Victor. Magic ink: Information software and the graphical interface. http://worrydream.com/MagicInk/, 2005.

[Wal02a] A. Walenstein. *Cognitive support in software engineering tools: A distributed cognition framework*. PhD thesis, Simon Fraser University, Vancouver, BC, 2002.

[Wal02b] Andrew Walenstein. Foundations of cognitive support: Toward abstract patterns of usefulness. In *Proceedings of the 14th Annual Conference on Design, Specification, and Verification of Interactive Systems*, Rostock, Germany, June 2002.

[Wal02c] Andrew Walenstein. Theory-based analysis of cognitive support in software comprehension tools. In *Proceedings of the 10th International Workshop on Program Comprehension*, pages 75–85, 2002.

[Wal03] Andrew Walenstein. Observing and measuring cognitive support: Steps toward systematic tool evaluation and engineering. In *IWPC '03: Proceedings of the 11th IEEE International Workshop on Program Comprehension*, pages 185–194, Washington, DC, USA, 2003. IEEE Computer Society.

[Wik] NCBO Wiki. Oboinowl:main page. http://www.bioontology.org/wiki/index.php?title=OboInOwl:Main_Page.

[wik06] Distributed cognition. http://en.wikipedia.org/wiki/Distributed_cognition, 2006.

[wor] Wordnet. http://wordnet.princeton.edu/.

[Yam07] Takashia Yamauchi. The semantic web and human inference: A lesson from cognitive science. In *Proceedings of ISWC/ASWC*, pages 609–622, November 2007.

[YF08] Takashi Yamauchi and Sean M. Falconer. Human factors for semantic web technologies. *International Journal of Human Computer Studies*, 2008. Under review.

[YM00] Takashi Yamauchi and Arthur B. Markman. Inference using categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26(3):776–795, 2000.

[Zhd] Anna V. Zhdanova. Owl ontology aligner. http://align.deri.org:8080/deri/align.jsp.

[Zhd05] Anna V. Zhdanova. Towards a community-driven ontology matching. In *K-CAP '05*, pages 221–222, 2005.

[ZN94] Jiajie Zhang and Donald A. Norman. Representations in distibuted cognitive tasks. *Cognitive Science*, 18:87–122, 1994.

Appendix A

Observational study: recruitment letter

FREE PIZZA

Participate in cutting-edge research at UVic!

Participate in a 1-1.5 hour study and interview where you work with a partner exploring two different software tools. No prior experience with the tools required. We are interested in how you learn and make decisions using the tools.

You will receive **free pizza** during the interview to thank you for your time.

To participate you need to:

1. Be experienced with computers and different software systems.
2. Be fluent in English.
3. Contact Sean Falconer by December 6th, 2006.

Location: The study will take place in the new Computer Science usability lab. **Contact:** Sean Falconer at seanf@uvic.ca or (250) 472-5778.

Appendix B

Observational study: obtaining informed consent

You are being invited to participate in a study entitled CogZ: Cognitive Support and Visualization for Human-Guided Mapping Systems that is being conducted by Sean Falconer.

Sean Falconer is a PhD Student in the department of Computer Science at the University of Victoria and you may contact him if you have further questions by telephone 250-472-5778 or email at seanf@uvic.ca.

As a graduate student, I am required to conduct research as part of the requirements for a degree in Doctorate of Philosphy. It is being conducted under the supervision of Dr. Margaret-Anne Storey. You may contact my supervisor at 250-472-5713 or by e-mail at mstorey@uvic.ca.

The purpose of this research project is to understand how users think and interact with mapping tools. Your responses will only be used for the purpose of this study.

Research of this type is important because mapping tools are widely used in both academia and industry. They require human interaction, however, little research has been conducted on helping the user interact and understand these systems. Understanding how to reduce the complexity of these tools is very important to reducing the cognitive load that users experience while working with these tools.

You are being asked to participate in this study because you have a significant amount of computer experience and you are proficient in the English language.

If you agree to voluntarily participate in this research, your participation will take approximately 90 minutes to complete. Participation involves filling out a short questionnaire related to your computer experience, working in a small group performing ontology/vocabulary alignments, and finally participating in a post-experiment interview where you will be encouraged to share your opinion and experience. These sessions will be recorded by video and audio. .

Participation in this study may cause some inconvenience to you since you must give up 90 minutes of your time to participate.

1.There are no known or anticipated risks to you by participating in this research.

The potential benefits of your participation in this research include benefiting society and the state of knowledge. This is the first research conducted to understand the cognitive processes involved while performing mappings between two ontologies. We feel that the requirements that will be developed as a result of this experiment will ultimately benefit ontology mapping users and researchers by making the tools easier to use.

As a way to compensate you for any inconvenience related to your participation, pizza will be provided during the post-experiment interview. It is important for you to know that it is unethical to provide undue compensation or inducements to research participants and, if you agree to be a participant in this study, this form of compensation to you must not be coercive. If you would not otherwise choose to participate if the compensation was not offered, then you should decline.

Your participation in this research must be completely voluntary. If you do decide to participate, you may withdraw at any time without any consequences or any explanation. If you do withdraw from the study your data will be destroyed and will not be used in any research related to this study. Furthermore, if you withdraw, pizza will still be available for you.

The researcher may have a relationship to potential participants as the researcher is a student in the same university as potential participants. To help prevent this relationship from influencing your decision to participate, the following steps to prevent coercion have been taken: general e-mail lists have been used for recruitment, general advertising within the CSC building, and finally, general recruitment from large CSC classes. No direct contact with potential participants will be conducted during recruitment.

In terms of protecting your anonymity, your name will never be associated with any research material generated from this study. All quotes will be anonymous. Given that participation will require working in groups of two, there are limitations to your confidentiality.

Your confidentiality and the confidentiality of the data will be protected by storying all hard-copy documents containing participant data in a locked filing cabinet, which only the principal investigator will have access to. Computerized information, such as the video and audio recordings, will be password protected and will be stored on a computer's hard drive in a locked office. At the end of the study, data will be burned to a CD or DVD and the originals will be deleted. These discs will also be stored in a locked filing cabinet.

It is anticipated that the results of this study will be shared with others in the following ways: computer science and software engineering journals, presented at scholarly meetings, and may form part of theses and dissertions.

Data from this study will be disposed of after a period of no longer than 3 years from the close of the study. All electronic data will be deleted. Any paper copies will be shredded, and any CDs or DVDs will be physically destroyed before being disposed of.

In addition to being able to contact the researcher or researchers supervisor at the above phone numbers, you may verify the ethical approval of this study, or raise any concerns you might have, by contacting the Associate Vice-President, Research at the University of Victoria (250-472-4545).

Your signature below indicates that you understand the above conditions of participation in this study and that you have had the opportunity to have your questions answered by the researchers.

Name of Participant:

Signature:

Date:

A copy of this consent will be left with you, and a copy will be taken by the researcher.

Appendix C

Observational study: pre-study questionnaire

1. What degree are you most likely to pursue (e.g. Computer Science)?
2. What year are you in (1, 2, 3, or 4)?
3. Approximately how many years have you been using computers?
4. How many hours a week do you use a computer?
5. What is your primary activity during your computer use (e.g. Web surfing, chatting, programming, etc.)?
6. Do you know what an ontology is (Yes or No)?
7. If yes, have you worked with ontology tools before (Yes or No)?
8. Have you ever performed a data alignment by hand or with a tool, for example between two databases, ontologies, XML schemas, etc.? If yes, please explain.

Appendix D

Survey study: mailing lists used

- dbworld - `http://www.cs.wisc.edu/dbworld`
- kaw@science.uva.nl
- semantic-web@w3.org
- kweb-all@lists.deri.org
- semanticweb@yahoogroups.com
- isworld@lyris.isworld.org
- seweb-list@lists.deri.org
- OM 2006 attendee mailing list

Appendix E

Survey study: recruitment letter

You are being invited to participate in a study entitled “CogZ : Cognitive Support and Visualization for Human-Guided Mapping Systems”. We are interested recruiting participants with experience performing ontology mappings. The study is being conducted by Sean Falconer, a PhD student in the department of Computer Science at the University of Victoria. You may contact Sean if you have questions. Phone (250) 474-5778 or e-mail seanf@uvic.ca

We are very interested in having you participate. Your participation is completely voluntary and you will not be identified by name in any work related to this study. Participation involves following the link provided below and filling out a short web-based survey related to your ontology mapping experience, tools that you use, types of ontologies that you use, and difficulties you have experienced.

Please contact Sean Falconer if you have further questions. If you would like to participate follow this link:

http:///link_to_study

Thank you,

Sean Falconer

PhD Student, Department of Computer Science

University of Victoria

The CHISEL Group

Appendix F

Survey study: obtaining information consent

Thank you for your interest in sharing your ontology mapping experience. Prior to completing this survey, the researchers involved and the Human Research Ethics Committee at the University of Victoria require that you be advised of the following:

- Your participation in this research is voluntary and anonymous.
- If you do decide to participate, you may withdraw at any time without any consequences or any explanation. If you decide to withdraw before submitting your responses, simply close the browser. Your responses will not be collected.
- Once you have submitted your questionnaire, it will be difficult to determine which one is yours. However, if you have left some uniquely identifying mark on it (for example, mistakenly putting your name on it), then contact Sean Falconer (seanf@uvic.ca). All efforts will be made to have your questionnaire deleted from the survey.
- There are no known or anticipated risks to you by participating in this study.
- The questionnaire process is completely anonymous. At no time will anyone be able to identify you with your responses. Your responses will only be used for the purpose of this study.
- Your responses are protected under the Freedom of Information and Protection of

Privacy Act. Study data will be kept for three years in secure machines. At the end of this time, the computer data files will be deleted.

- Results from this questionnaire will be published in computer science and software engineering journals, presented at scholarly meetings, and will form part of several thesis and dissertations. Results will only ever be presented in aggregate form.
- If you need help filling out the survey, send an e-mail to Sean Falconer seanf@uvic.ca or telephone 250-472-5778.
- Please contact Dr. Margret-Anne Storey or Mr. Sean Falconer if you have further questions. Phone: 250-472-5713 or email Dr. Storey: mstorey@cs.uvic.ca, Mr. Falconer: 250-472-5778 or seanf@uvic.ca. For more information on our research group, look at http://www.thechiselgroup.org
- In addition to being able to contact the study researchers, you may verify the ethical approval of this study, or raise any concerns you might have, by contacting the Associate Vice-President, Research at the University of Victoria (250-472-4362).

You must read and accept the above conditions before completing the survey.

I have read the above conditions and accept them. YES NO

Appendix G

Interview study: recruitment letter

To whom it may concern,

My name is Sean Falconer and I am a PhD candidate from the Chisel Lab at the University of Victoria. I am currently leading a user study aimed at understanding the complexity of ontology mapping, the problems users experience, and how tools can be better designed to support users during the mapping procedure.

I am recruiting subjects working on mapping projects to participate in a research study. Participants of this study will be asked to take part in an interview session where I will ask them questions regarding their experience working on mapping projects and performing mapping related tasks.

Members of your research group/department would be ideal candidates. It is estimated that the interview will take no more than one hour.

If you are interested, please forward this letter onto your team members and have them contact me directly to arrange an interview time. We have been ethically approved by our local ethics board.

Thank you,

Sean Falconer

250-472-5778

seanf@uvic.ca

Appendix H

Interview study: obtaining informed consent

You are being invited to participate in a study entitled CogZ: Cognitive Support and Visualization for Human-Guided Mapping Systems that is being conducted by Sean Falconer.

Sean Falconer is a PhD Candidate in the department of Computer Science at the University of Victoria and you may contact him if you have further questions by telephone 250-472-5778 or email at seanf@uvic.ca.

As a graduate student, I am required to conduct research as part of the requirements for a degree in Doctorate of Philosophy. It is being conducted under the supervision of Dr. Margaret-Anne Storey. You may contact my supervisor at 250-472-5713 or by e-mail at mstorey@uvic.ca.

The purpose of this research project is to understand how users think and carry out mappings as well as interact with mapping tools. Your responses will only be used for the purpose of this study.

Research of this type is important because mapping tools are widely used in both academia and industry. They require human interaction, however, little research has been conducted on helping the user interact and understand these systems. Understanding how to reduce the complexity of these tools is very important to reducing the cognitive load that users experience while working with these tools.

You are being asked to participate in this study because you have a significant amount of ontology mapping experience. If you agree to voluntarily participate in this research, your participation will take approximately 1 hour to complete. Participation involves allowing yourself to be interviewed about your mapping experience. The audio recordings will only be used for analysis of these sessions and no one besides the investigators on this project will hear these recordings.

Participation in this study may cause some inconvenience to you since you must give up 1 hour of your time to participate. There are no known or anticipated risks to you by participating in this research.

The potential benefits of your participation in this research include benefiting society and the state of knowledge. This is one of the first research studies aimed at understanding the user process involved in ontology mapping. We feel that the requirements that will be developed as a result of this experiment will ultimately benefit ontology mapping users and researchers by making the tools easier to use.

Your participation in this research must be completely voluntary. If you do decide to participate, you may withdraw at any time without any consequences or any explanation. If you do withdraw from the study your data will be destroyed and will not be used in any research related to this study. The researcher may have a relationship to potential participants as the researcher is working the area of ontology mapping and may have had contact with potential participants. To help prevent this relationship from influencing your decision to participate, the following steps to prevent coercion have been taken: general e-mail lists have been used for recruitment. No direct contact with potential participants will be conducted during recruitment.

In terms of protecting your anonymity, your name will never be associated with any research material generated from this study. All quotes will be anonymous. Given that participation will require you to participate in a focus group with other members of your mapping project team, there are limitations to your confidentiality. Your confidentiality and the confidentiality of the data will be protected by storing all hard-copy documents containing participant data in a locked filing cabinet, which only the principal investigator will have access to. Computerized information, such as the audio recordings, will be password protected and will be stored on a computer's hard drive in a locked office. At the end of the study, data will be burned to a CD or DVD and the originals will be deleted. These discs will also be stored in a locked filing cabinet.

It is anticipated that the results of this study will be shared with others in the following ways: computer science and software engineering journals, presented at scholarly meetings, and may form part of theses and dissertations.

Data from this study will be disposed of after a period of no longer than 3 years from the close of the study. All electronic data will be deleted. Any paper copies will be shredded, and any CDs or DVDs will be physically destroyed before being disposed of.

In addition to being able to contact the researcher or researchers supervisor at the above phone numbers, you may verify the ethical approval of this study, or raise any concerns you might have, by contacting the Associate Vice-President, Research at the University of Victoria (250-472-4545).

Your signature below indicates that you understand the above conditions of participation in this study and that you have had the opportunity to have your questions answered by the researchers.

Name of Participant:

Signature:

Date:

A copy of this consent will be left with you, and a copy will be taken by the researcher.

Appendix I

Interview study: example of interview codes

- Background/scale of project
  - U = use case
  - Sc = scale
  - OW = other similar work to use case
- Process
  - Sim = simplifying
  - Def = problem definition
  - MM = mapping methodology
  - MP = mapping presentation
  - MR = mapping review
  - MT = mapping tracking (saving mappings, flagging mapping cases)
  - D = difficulties
- Team
  - TC = team construction
  - TM = team methodology

  - TR = team results review

- Tools

  - TA = tools available
  - TU = tools used/not used
  - TL = tool limitations
  - TFR = tool future requirements

Appendix J

Evaluation study: recruitment letter

FREE iTunes Gift Card/Movie Tickets/$20 Cash

Participate in cutting-edge research at UVic!

Looking for experienced computer users to take part in user study.

Participate in a 1.5 - 2 hour study where you will carry out tasks with two software tools. No prior experience with the tools required.

You will receive either a $20 iTunes card, two free movie tickets, or $20 cash to thank you for your time.

To participate you need to:

1. Be experienced with computers and different software systems.
2. Be fluent in English.
3. Contact Sean Falconer.

Location: The study will take place in the new Computer Science usability lab. **Contact:** Sean Falconer at seanf@uvic.ca or (250) 472-5778.

Appendix K

Evaluation study: obtaining informed consent

You are being invited to participate in a study entitled CogZ: Cognitive Support and Visualization for Human-Guided Mapping Systems that is being conducted by Sean Falconer.

Sean Falconer is a PhD Candidate in the department of Computer Science at the University of Victoria and you may contact him if you have further questions by telephone 250-472-5778 or email at seanf@uvic.ca.

As a graduate student, I am required to conduct research as part of the requirements for a degree in Doctorate of Philosophy. It is being conducted under the supervision of Dr. Margaret-Anne Storey and in collaboration with Dr. Takashi Yamauchi at Texas A&M University. You may contact my supervisor at 250-472-5713 or by e-mail at mstorey@uvic.ca.

The purpose of this research project is to help evaluate an ontology mapping tool I have been developing as part of my research. Your responses will only be used for the purpose of this study.

Research of this type is important because mapping tools are widely used in both academia and industry. They require human interaction, however, little research has been conducted on helping the user interact and understand these systems. Understanding how to reduce the complexity of these tools is very important to reducing the cognitive load that users experience while working with these tools.

If you agree to voluntarily participate in this research, your participation will take approximately 1-2 hours to complete. Participation involves filling out a short pre-study questionnaire, a short training session, carrying out some predefined tasks with the tools, and finally filling out a short usability survey. These sessions will be recorded by video and audio. The audio and video recordings will only be used for analysis of these sessions and no one besides the investigators on this project will view these recordings.

Participation in this study may cause some inconvenience to you since you must give up 1-2 hours of your time to participate. There are no known or anticipated risks to you by participating in this research.

The potential benefits of your participation in this research include benefiting society and the state of knowledge. This is will be the first study conducted to evaluate the usability of the two tools. We feel that this evaluation will help further improve the tools and contribute to the overall requirements needed for making the tools easier to use.

As a way to compensate you for any inconvenience related to your participation, you will receive either a $20 iTunes gift card or two free movie tickets. It is important for you to know that it is unethical to provide undue compensation or inducements to research participants and, if you agree to be a participant in this study, this form of compensation to you must not be coercive. If you would not otherwise choose to participate if the compensation was not offered, then you should decline. Your participation in this research must be completely voluntary. If you do decide to participate, you may withdraw at any time without any consequences or any explanation. If you do withdraw from the study your data will be destroyed and will not be used in any research related to this study.

The researcher may have a relationship to potential participants as the researcher is a student at the University of Victoria and may have had contact with potential participants. To help prevent this relationship from influencing your decision to participate, the following steps to prevent coercion have been taken: general e-mail lists have been used for recruitment, general advertising within the CSC building, and finally, general recruitment from large CSC classes. No direct contact with potential participants will be conducted during recruitment.

In terms of protecting your anonymity, your name will never be associated with any research material generated from this study. All quotes will be anonymous. Your confidentiality and the confidentiality of the data will be protected by storing all hard-copy documents containing participant data in a locked filing cabinet, which only the principal investigator will have access to. Computerized information, such as the video and audio recordings, will be password protected and will be stored on a computer's hard drive in a locked office. At the end of the study, data will be burned to a CD or DVD and the originals will be deleted. These discs will also be stored in a locked filing cabinet.

It is anticipated that the results of this study will be shared with others in the following ways: computer science and software engineering journals, presented at scholarly meetings, and may form part of theses and dissertations.

Data from this study will be disposed of after a period of no longer than 3 years from the close of the study. All electronic data will be deleted. Any paper copies will be shredded, and any CDs or DVDs will be physically destroyed before being disposed of.

In addition to being able to contact the researcher or researchers supervisor at the above phone numbers, you may verify the ethical approval of this study, or raise any concerns you might have, by contacting the Associate Vice-President, Research at the University of Victoria (250-472-4545).

Your signature below indicates that you understand the above conditions of participation in this study and that you have had the opportunity to have your questions answered by the researchers.

Name of Participant:

Signature:

Date:

A copy of this consent will be left with you, and a copy will be taken by the researcher.

Appendix L

Evaluation study: CogZ experimenter handbook

1. Sign consent
    - Have the participant read and sign the consent form.
2. Explain study
    - Explain the purpose of the study.
    - Explain what an ontology is.
    - Explain what mapping means.
3. Training with tool CogZ
    - Open Prompt with the CMU and UMD ontologies.
    - Explain the ontologies.
    - Explain CogZ's interface.
        - Source tree
        - Target tree
        - Mappings represented visually
        - Neighborhood view
        - Suggestion list

    - Show how to get concept details
    - Show search
    - Show filters
  - Have the participant accept the suggested mapping from Thesis to Thesis.
  - Have the participant remove the suggested mapping between Publication and Location.
  - Have the user create a user-defined mapping between Sex and Gender.
  - Show how to get the list of completed mappings.
  - Give participant CogZ help sheet.

4. Study tasks with CogZ
   - Explain that we are going to move onto the study tasks.
   - Open CogZ with sportEvent and sportSoccer ontologies.
   - Explain to the user what these ontologies represent.
   - Give the participant the list of tasks and tell them they have 30 minutes to do as much as they can.
   - Encourage them to think aloud.
   - Again explain that they can ask questions but that they should attempt to use the help sheet.
5. SUS questionnaire for CogZ
   - Give questionnaire to participant.
6. 10 minute break
7. Training with Prompt
   - Explain Prompt's interface.
     - Source tree

    - Target tree
    - Mappings represented visually
    - Neighborhood view
    - Suggestion list
    - Show how to get concept details
    - Show search
    - Show filters
  - Have the participant accept the suggested mapping from Thesis to Thesis.
  - Have the participant remove the suggested mapping between Publication and Location.
  - Have the user create a user-defined mapping between Sex and Gender.
  - Show how to get the list of completed mappings.
  - Give participant Prompt help sheet.

8. Study tasks with Prompt
   - Explain that we are going to move onto the study tasks.
   - Open Prompt with sportEvent and sportSoccer ontologies.
   - Explain to the user what these ontologies represent.
   - Give the participant the list of tasks and tell them they have 30 minutes to do as much as they can.
   - Encourage them to think aloud.
   - Again explain that they can ask questions but that they should attempt to use the help sheet.
9. SUS questionnaire for Prompt
   - Give questionnaire to participant.
10. Conduct interview

Appendix M

Evaluation study: Prompt experimenter handbook

1. Sign consent
   - Have the participant read and sign the consent form.
2. Explain study
   - Explain the purpose of the study.
   - Explain what an ontology is.
   - Explain what mapping means.
3. Training with Prompt
   - Open Prompt with the CMU and UMD ontologies.
   - Explain the ontologies.
   - Explain Prompt's interface.
     - Source tree
     - Target tree
     - Mappings represented visually
     - Neighborhood view
     - Suggestion list

  - Show how to get concept details
  - Show search
  - Show filters
- Have the participant accept the suggested mapping from Thesis to Thesis.
- Have the participant remove the suggested mapping between Publication and Location.
- Have the user create a user-defined mapping between Sex and Gender.
- Show how to get the list of completed mappings.
- Give participant Prompt help sheet.

4. Study tasks with Prompt
   - Explain that we are going to move onto the study tasks.
   - Open Prompt with sportEvent and sportSoccer ontologies.
   - Explain to the user what these ontologies represent.
   - Give the participant the list of tasks and tell them they have 30 minutes to do as much as they can.
   - Encourage them to think aloud.
   - Again explain that they can ask questions but that they should attempt to use the help sheet.
5. SUS questionnaire for Prompt
   - Give questionnaire to participant.
6. 10 minute break
7. Training with tool CogZ
   - Explain CogZ's interface.
     - Source tree

  - Target tree
  - Mappings represented visually
  - Neighborhood view
  - Suggestion list
  - Show how to get concept details
  - Show search
  - Show filters

- Have the participant accept the suggested mapping from Thesis to Thesis.
- Have the participant remove the suggested mapping between Publication and Location.
- Have the user create a user-defined mapping between Sex and Gender.
- Show how to get the list of completed mappings.
- Give participant CogZ help sheet.

8. Study tasks with CogZ
   - Explain that we are going to move onto the study tasks.
   - Open CogZ with sportEvent and sportSoccer ontologies.
   - Explain to the user what these ontologies represent.
   - Give the participant the list of tasks and tell them they have 30 minutes to do as much as they can.
   - Encourage them to think aloud.
   - Again explain that they can ask questions but that they should attempt to use the help sheet.
9. SUS questionnaire for CogZ
   - Give questionnaire to participant.
10. Conduct interview

Appendix N

Evaluation study: System Usability Scale (SUS) questionnaire

1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in the system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Appendix O

Evaluation study: example of interview codes

C = confusion	D = duplication
CL = clusers	VM = view mappings
N = navigation	DD = drag and drop
F = filtering	CS = context switch
VR = view relationships	A = annoyance
LC = learning curve	CF = confidence
P = productivity	E = easier
CT = context	IS = ignored suggestions
MH = multiple hierarchies	P = problem
S = search	U = usability issue