Linköping Studies in Science and Technology

Thesis No. 1468

Dealing with Missing Mappings and Structure in a Network of Ontologies

by

Qiang Liu

Submitted to Linköping Institute of Technology at Linköping University in partial fulfilment of the requirements for degree of Licentiate of Engineering

Department of Computer and Information Science
Linköping universitet
SE-581 83 Linköping, Sweden

Linköping 2011

ISBN 978-91-7393-233-2, ISSN 0280–7971
Printed by LiU-Tryck, 2011

Electronic version available at :
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-64281

Dealing with Missing Mappings and Structure in a Network of Ontologies

by

Qiang Liu

March 2011
ISBN 978-91-7393-233-2
Linköping Studies in Science and Technology
Thesis No. 1468
ISSN 0280–7971
LiU–Tek–Lic–2011:7

ABSTRACT

With the popularity of the World Wide Web, a large amount of data is generated and made available through the Internet everyday. To integrate and query this huge amount of heterogeneous data, the vision of Semantic Web has been recognized as a possible solution. One key technology for the Semantic Web is ontologies. Many ontologies have been developed in recent years. Meanwhile, due to the demand of applications using multiple ontologies, mappings between entities of these ontologies are generated as well, which leads to the generation of ontology networks consisting of ontologies and mappings between these ontologies. However, neither developing ontologies nor finding mappings between ontologies is an easy task. It may happen that the ontologies are not consistent or complete, or the mappings between these ontologies are not correct or complete, or the resulting ontology network is not consistent. This may lead to problems when they are used in semantically-enabled applications. In this thesis, we address two issues relevant to the quality of the mappings and the structure in the ontology network. The first issue deals with the missing mappings between networked ontologies. Assuming existing mappings between ontologies are correct, we investigate whether and how to use these existing mappings, to find more mappings between ontologies. We propose and test several strategies of using the given correct mappings to align ontologies. The second issue deals with the missing structure, in particular missing is-a relations, in networked ontologies. Based on the assumption that missing is-a relations are a kind of modeling defects, we propose an ontology debugging approach to tackle this issue. We develop an algorithm for detecting missing is-a relations in ontologies, as well as algorithms which assist the user in repairing by generating and recommending possible ways of repairing and executing the repairing. Based on this approach, we develop a system and test its use and performance.

This work has been supported by CUGS (the Swedish National Graduate School in Computer Science).

Department of Computer and Information Science
Linköping universitet
SE-581 83 Linköping, Sweden

Acknowledgements

It is my great pleasure to express my appreciation to those who have helped me with this thesis. Without them, this thesis would not have been possible.

Firstly, I would like to thank my supervisor Professor Patrick Lambrix, who introduced me into this research area and kept encouraging and supporting my work during these years. His rigorous scholarship, patient guidance and insightful thoughts have left deep impression on me. The constructive discussions and conversations with him build up the ground of the work in this thesis. I am grateful to my co-supervisor Professor Nahid Shahmehri for giving valuable comments on my research career and offering such a nice work environment. I would also like to thank the other two co-supervisors Professor Henrik Eriksson and Doctor He Tan for providing constructive suggestions on my research work.

I would also like to show my gratitude to all the present and former colleagues at the Division for Database and Information Techniques (ADIT) for the sharing of their experiences and joy with me. Special thanks to Anna Vapen for her kind help in daily life, as well as Doctor Jose M. Penã and Associate Professor Lena Strömbäck for their assistance on my teaching.

I acknowledge the financial support of the Swedish National Graduate School in Computer Science (CUGS).

I feel grateful to my dear parents, who have always been a great support and encouragement for me. Last but not least, I would like to express my deepest gratitude to my beloved wife, Yaqian, for accompanying me through all the difficult times and comforting my soul with her undemanding love and enduring patience.

Qiang, Liu
Linköping, Sweden
December, 2010

List of Primary Publications

This thesis is mainly based on the following papers.

1. Lambrix P, Liu Q, Debugging the missing is-a structure within ontologies networked by partial reference alignments, submitted.

2. Lambrix P, Liu Q, Tan H, Repairing the missing is-a structure of ontologies, In *Proceedings of the 4th Asian Semantic Web Conference - ASWC09*, LNCS 5926, pp 76-90, Shanghai, China, 2009.

3. Lambrix P, Liu Q, Using partial reference alignments to align ontologies, In *Proceedings of the 6th European Semantic Web Conference - ESWC09*, LNCS 5554, pp 188-202, Heraklion, Greece, 2009.

List of Other Publications

In addition to the papers listed above, there are the following related publications.

1. Liu Q, Lambrix P, A system for debugging missing is-a structure in networked ontologies, In *Proceedings of the 7th International Conference on Data Integration in the Life Sciences - DILS 2010*, LNBI 6254, pp 50-57, Gothenburg, Sweden, 2010.

2. Liu Q, Lambrix P, Debugging the missing is-a structure of networked ontologies, In *Proceedings of the 7th Extended Semantic Web Conference - ESWC10*, Part II, LNCS 6089, pp 478-482, Heraklion, Greece, 2010.

3. Lambrix P, Liu Q, Tan H, RepOSE: an environment for repairing missing ontological structure, In *Proceedings of the 4th Asian Semantic Web Conference (Demos) - ASWC09*, LNCS 5926, pp 365-366, Shanghai, China, 2009.

4. Lambrix P, Liu Q, Tan H, A system for repairing missing is-a structure in ontologies, In *Semantic Web Applications and Tools for Life Sciences (Demos) - SWAT4LS09*, CEUR 559, Amsterdam, Netherlands, 2009.

5. Lambrix P, Liu Q, Tan H, Aligning Anatomy Ontologies in the Ontology Alignment Evaluation Initiative, In *the 25th Workshop of the Swedish Artificial Intelligence Society*, pp 13-20, Linköping, Sweden, 2009.

6. Lambrix P, Tan H, Liu Q, SAMBO and SAMBOdtf results for the Ontology Alignment Evaluation Initiative 2008, In *Proceedings of the 3rd International Workshop on Ontology Matching*, CEUR 431, pp 190-198, Karlsruhe, Germany, 2008.

Contents

Chapter 1

Introduction

1.1 Motivation

With the popularity of the World Wide Web (WWW), a large amount of data is generated and made available through the Internet everyday. According to the statistics[1], WWW currently contains at least 2.54 billion pages accessed by millions of users. Although this data is extremely useful, we are not using its full capacity. The data on the WWW is heterogeneous in different ways, stored behind various applications, and also updated frequently. It is hard to integrate and query this huge amount of data. In particular, the contents of most information are designed to be read by people (e.g. in HTML format), not machines. In the case that intelligent agents could not understand the meaning, or the "semantics", of the information, a lot of tasks still largely need the direction of humans. To alleviate these difficulties, researchers have proposed the vision of the Semantic Web. The primary goal of the Semantic Web is to add semantics to the current web, by explicitly describing and relating objects using formal representations that machines can understand and process [1]. It provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [2]. At present, the Semantic Web has been referred as the future of current WWW by the World Wide Web Consortium (W3C), and its technologies have been applied in many areas, such as knowledge management, electronic commerce and bioinformatics [3, 4, 5].

As a key Semantic Web technology, ontologies play an important role in laying the ground for semantic interoperability. Ontologies can be seen as defining the basic terms and relations of a domain of interest, as well as the rules for combining these terms and relations [6]. With data annotated and indexed by terms from the same ontology, the meaning of the content in diverse information sources becomes explicit and machine-understandable, so that communication between people and application systems could be done on a shared and common understanding of a domain. The benefits of the use of ontologies include information reuse, sharing and

[1] http://www.worldwidewebsize.com/

portability across platforms, as well as improved documentation, maintenance, reliability and interoperability between applications [7]. With the significance of ontologies being widely recognized by research communities and companies, many ontologies have been developed in recent years in many areas [8].

With different communities and companies involved in the development of ontologies, it becomes a common situation that multiple ontologies co-exist in the same area for similar application purposes. These ontologies are defined from different views, represented in different languages and using different terminologies, which poses as a barrier to semantic interoperability on the ontology level. In response, *ontology alignment*, which finds the mappings between semantically related entities of different ontologies, provides a promising solution [9]. With ontologies being aligned with each other, the relationships between different ontologies are explicit, which enables semantic applications using different ontologies to communicate with each other or to deal with multiple ontologies. Ontology alignment leads to improvements in search, integration and analysis of data and it is considered as one of the substantial research issues for the further development and success of the Semantic Web [8]. Hence, in parallel with the development of ontologies, researchers put great efforts on building ontology alignment systems and generating mappings for the ontologies (e.g. overview in [10]).

With more and more ontologies and mappings developed for the public use, a number of systems and portals have been set up for storing ontologies and the mappings between them. In the field of bioinformatics, for instance, there are BioPortal[2] and Unified Medical Language System (UMLS)[3], which not only serve as repositories for ontologies and mappings, but also offer support for international research cooperation on the development of ontologies and mappings, and the use of them for the Semantic Web. During this process, several *ontology networks*, consisting of ontologies and mappings between these ontologies, come into sight.

In practice, neither developing ontologies nor finding mappings between ontologies is an easy task. It may happen that the resulting ontologies are not consistent or complete, or the mappings between these ontologies are not correct or complete, or the ontology network consisting of ontologies and mappings is not consistent. This may lead to problems when these ontologies or mappings are used in semantically-enabled applications. Wrong conclusions may be derived or valid conclusions may be missed. For instance, the missing mappings between ontologies influence the ontology-based database integration, where the information in different databases is annotated with terms in different ontologies and intended to be united for query answering. The missing mappings between terms across ontologies are likely to cause missed relations between related information across databases, affecting the quality of query results. In a similar manner, the missing structural relations in ontologies can influence the ontology-based literature search, where queries are refined and expanded by moving up and down the hierarchy of concepts in the background ontology. As an example, suppose we want to find

[2]http://bioportal.bioontology.org/

[3]http://www.nlm.nih.gov/research/umls/about_umls.html

articles in MeSH (Medical Subject Headings[4], an ontology represented as the controlled vocabulary of the U.S. National Library of Medicine) Database of PubMed[5] using the term *Scleral Diseases* in MeSH. By default the query will follow the hierarchy of MeSH and include more specific terms for searching, such as *Scleritis*. If the relation between *Scleral Diseases* and *Scleritis* is missing in MeSH, we will miss 738 articles (about 55% of the original result) in the search result. Besides, it is well-known that people that are not expert in knowledge representation often misuse and confuse equivalence, is-a and part-of (e.g. [11]), which leads to problems in the structure of the ontologies.

1.2 Problem Statement

In this thesis, we address two problems relevant to the ontology network, the missing mappings between networked ontologies and the missing structure in networked ontologies. For both problems, we assume that the existing mappings between every pair of ontologies in the network are correct, such that they constitute a *Partial Reference Alignment (PRA)* between the two related ontologies. A PRA is defined as a set of correct mappings between a pair of ontologies.

1. For the first problem, the missing mappings between networked ontologies, we formulated it as an ontology alignment task as follows. **Given two ontologies networked by a PRA, is it possible, and if so, how to use the PRA to find more correct mappings between these two ontologies?** In our work, we only consider mappings which are *equivalence* relations between *concepts* of different ontologies, since they are the most common results from current ontology alignment systems [12]. Our study is conducted in a general ontology alignment framework [13], which is applicable for most existing ontology alignment systems.

2. The second problem that we tackle in this thesis is the missing structure in networked ontologies. Especially, we focus on the missing is-a relations in ontologies networked by PRAs. Considering the missing is-a relations in ontologies as one kind of modeling defects, we deal with the following problem. **Given a set of ontologies networked by PRAs, how to detect and repair the missing is-a relations in these networked ontologies?** In this work, we only consider mappings which are *equivalence*, *subsumed-by* and *subsumes* relations between *concepts* of different ontologies. Besides, we assume that all the existing is-a relations in the ontologies are correct. By this means, our approach uses the ontology network as domain knowledge to detect the missing is-a relations in these ontologies. It further assists the user in repairing the ontologies by generating and recommending possible ways of repairing and executing the repairing.

[4] http://www.nlm.nih.gov/mesh/
[5] http://www.ncbi.nlm.nih.gov/pubmed/

1.3 Contributions

The main contributions of the work presented in this thesis are as follows.

1. Regarding the first problem

    - We developed several approaches for using PRAs in the ontology alignment process. We developed algorithms for using PRAs to partition ontologies, to compute similarities between terms and to filter mapping suggestions.
    - We did extensive experimental study involving most components of an ontology alignment system. The results show that PRAs can be used for aligning ontologies and give indications about which methods should be used.
    - Even though there are many existing ontology alignment systems and since 2008 there have been calls for the use of PRAs for aligning ontologies from the community[6], we are the first to study and experiment with the use of PRAs in ontology alignment.

2. Regarding the second problem

    - We proposed an ontology debugging approach to detect and repair the missing is-a structure in ontologies networked by PRAs. Even though the is-a relation is a fundamental and the most commonly used relation in ontologies, there is not much work on debugging missing is-a structure in ontologies, and even fewer in the context of networked ontologies.
    - We proposed a new approach for repairing missing is-a relations which takes the preferences of domain experts into account, rather than trivially adding the missing is-a relations to the relevant ontology. Further, we developed algorithms to generate and recommend possible ways of repairing, as well as algorithms for ranking missing is-a relations and executing the repairing.
    - We developed a system, which allows a domain expert to debug the missing is-a structure of networked ontologies in a semi-automatic way.

1.4 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 presents the background of this thesis. The basic concepts about ontologies are introduced, as well as some Semantic Web ontologies in applications nowadays. The research fields of ontology alignment and ontology debugging are generally described afterwards.

[6]In Ontology Alignment Evaluation Initiative (OAEI), a yearly international competition for evaluating ontology alignment systems, the problem of using PRAs to align ontologies has been proposed as a task in Anatomy track ever since 2008.

Chapter 3 presents our work (published in [14]) on missing mappings between networked ontologies, which is, in particular, the use of partial reference alignments to align ontologies. Several approaches are presented for using PRAs in different components of an ontology alignment system, with experiments and results discussed regarding the effects of these approaches.

In Chapter 4, we present our work (submitted [15] and partially published in [16]) on missing structure (is-a hierarchy) in networked ontologies. Considering the missing is-a structure as a kind of modeling defects in ontologies, we propose an ontology debugging approach to detect and repair these missing is-a relations. A developed system based on this approach and its use are described, with experiments discussed afterwards.

Finally, Chapter 5 concludes the thesis and discusses the possible directions for the future work.

Chapter 2

Background

In this chapter, the basic concepts of ontologies are presented in the context of the Semantic Web, as well as some developed ontologies for the real life applications. Thereafter, the research fields of ontology alignment and ontology debugging are briefly introduced.

2.1 Semantic Web and Ontologies

The vision of the Semantic Web is initiated by the Tim Berners-Lee, the inventor of the WWW. In his book [1], he describes the Semantic Web as the future of the current WWW, where information is represented as a web of data in the machine-readable form with well-defined meaning, so that software agents can understand, process and communicate the information automatically to accomplish the sophisticated tasks for users. It will change the industries and our experience of internet [17]. To reach this goal, a necessary step is to structure the information of web pages in a standard way, with inherent knowledge understandable, processable and exchangeable by the computer programs. Ontologies are recognized as a good solution for this purpose.

2.1.1 What is an Ontology?

As a term stemming from philosophy, ontology is originally defined as the study of being, the study of existence. It tries to answer the questions such as "what is the nature of being" and "what are the features common to all beings" [18]. In the middle of 1970s, researchers from the Artificial Intelligence (AI) field have recognized that capturing knowledge is the key to build large and powerful AI systems [6], and ontologies began drawing their attention as effective computational models to build shared, reusable knowledge bases processable by the computer programs. In this context, a pragmatic definition of the ontology given by Neches et al. is : "*An ontology defines the basic terms and relations comprising the vocabulary of a topic* *area as well as the rules for combining terms and relations to define extensions to the vocabulary*" [6], which points out the way of building an ontology.

In practice, AI researchers use ontologies in different ways for different purposes, which derives many other definitions of ontology appearing in the literature within this field. Some researchers define the ontology as a specific syntactic object, such as representation of conceptual system via a logical theory, the vocabulary used by a logical theory or a meta-level specification of a logical theory. Some define the ontology as a conceptual semantic entity, for instance, informal conceptual system or formal semantic account. For an overview and discussion about these different interpretations, we refer to [18]. One definition that has been referenced a lot is given by Gruber, which describes an ontology as "an *explicit specification of a conceptualization*", where a conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose [19]. According to this definition, an ontology contains definitions of the concepts, relationships, and other distinctions that are relevant for modeling a domain, as well as formal axioms that constrain the interpretation and well-formed use of these terms. It is a broad definition in the sense that the interpretation allows a range of specifications from simple glossaries to logical theories [20].

As we can see, ontology is a rather high-level concept. The interpretation and design of ontologies differs with the kind of information they represent as well as the way they are used.

2.1.2 Knowledge Representation for Ontologies

Even though ontologies differ regarding the kind of information they can represent, most ontologies contain the following components in its knowledge representation [21] [22].

- **Concepts** represent sets or classes of entities within a domain. For example, "Protein" is a concept in the molecular biology domain, representing all the things that are proteins.

- **Instances** are the entities represented by a concept. However, since ontology is supposed to be a conceptualization of the domain, instances are usually not included.

- **Relations** describe the ways in which concepts and instances are related to one another. There are many types of relations. The most common two are subsumption relationships (e.g. "Protein" *is-a* "Organic Compound") and the partitive relationships (e.g. "Nucleus" *is-part-of* "Cell"), which are widely used to organize concepts into taxonomies.

- **Axioms** describe facts that are always true in the topic area of the ontology (e.g. every "Cell" has a "Cell Membrane"), which can be used to constrain values for concepts or instances.

Depending on which of the components are included in the knowledge representation, ontologies can be categorized into different kinds. A simple type of ontology is *controlled vocabulary*, which is in the form of list of concepts each of which has an explicit and non-redundant definition. When concepts from a collection of controlled vocabulary are organized into a hierarchy structure we obtain a *taxonomy*, where the hierarchy is usually made of subsumption or partitive relationships. *Thesaurus* is a more complex kind of ontology, where concepts from a collection of controlled vocabulary are networked through a fixed set of relations based on standards [23], such as synonym, broader/narrower term and associated/related term. An extension of thesaurus is *data models*, which further allow for defining attributes (which are the properties that the concepts can have, also called functions in [21]) and a limited form of axioms. The most expressive kind of ontology is *knowledge base*, which is usually based on a logic. It can represent all types of components in a logically consistent manner, so that classical deduction can be used to reason about the knowledge.

Once the components are decided for building an ontology of a specific kind, we must choose the appropriate language with certain degree of formality to represent the ontology. The formality of ontology languages ranges from very informal to strictly formal. In [24], Uschold and Gruninger identified four kinds of formalities, which include *highly informal* if the expression uses loose natural language, *structured-informal* if the expression is in a explicit, restricted and structured form of natural language, *semi-formal* if the expression is based on formally defined languages, and *rigorously formal* if the expression has formal semantics, theorems and proofs of such properties as soundness and completeness.

Compared with informal representation language, using formal representation language is likely to reduce ambiguity in the ontology and improve interoperation. Having a well-defined semantics (usually based on logic theory), the formal language allows ontologies to provide automatic reasoning services, such as deriving conclusions from statements in the ontologies or build the subsumption hierarchy. However, building ontologies using formal languages is a challenging task. "If there are no sufficient axioms, using a formal language may be as or more ambiguous than a detailed carefully crafted set of definitions in natural language" [25]. Besides, using less formal language to develop ontologies is usually easier and faster to have results mature for use in a working system. Therefore, in practice, many ontologies start out as controlled vocabulary, with more advanced representation and functionalities gradually furnished afterwards [10].

2.1.3 Ontology Languages for the Semantic Web

The basic language to describe resources on the Semantic Web is Resource Description Framework (RDF). It uses a simple assertional language in the form of triples, allowing the integration of data from different kinds of resources [26]. In RDF, every triple is a statement expressed as "subject-predicate-object", which is used to specify the value of a property belonging to a subject. In parallel with RDF, RDF vocabulary description language, called RDF Schema (RDFS), extends RDF vocabulary with functions to describe the taxonomies of concepts and properties [27]. It also introduces some constraints on the properties, such as domain and range. However, RDF/RDFS is still a limited ontology language, unable to define, for example, cardinality constraints and conjunction of concepts.

In need of more expressive ontology languages, the Semantic Web research communities have proposed several Web ontology languages, including Simple HTML Ontology Extensions (SHOE) [28], Ontology Inference Layer (OIL) [29] and DAML+OIL [30]. Based on these efforts, W3C proposed the standard Web Ontology Language (OWL) [31] in 2004, which is now a broadly accepted ontology language for the Semantic Web. With a well defined syntax for easy data exchange, OWL is based on the earlier work of OIL and DAML+OIL, while (partially) compatible with the RDF and RDFS. Compared with RDF/RDFS, OWL provides a richer set of language primitives to support more expressive representations, such as the disjointness of concepts, logic combinations of concepts, cardinality restrictions on the properties and some special characteristics of properties (e.g. a property is transitive or unique or inverse of another property) [32]. One of the main benefits of OWL is that it has a formal semantics based on predicate logic, which brings the power of automatic reasoning to the Semantic Web.

In spite of the benefits, using an expressive language is usually at the cost of inefficient reasoning, which may lead to computational intractability. To deal with this, OWL is further subdivided into three species – *OWL Lite*, *OWL DL* and *OWL FULL*. OWL Lite is the simplest OWL language, which can be used to express taxonomies and simple constraints. OWL DL is based on a subset of Description Logic (DL) *SHIQ* [33], which retains the logic decidability together with efficient reasoning. OWL Full allows all languages primitives, having no constraints for expressiveness. However, OWL Full is too expressive such that the reasoning in it becomes undecidable [34].

2.1.4 Semantic Web Ontologies in Applications

With the development of the Semantic Web, many ontologies have been generated for semantically-enabled applications. Most of them could be classified into two categories, *upper ontologies* (or foundation ontologies) and *domain ontologies* [35].

Upper ontologies serve to provide a high level model by formalizing common notions applicable across a broad range of domains, such as processes and events, time and space, roles and functions [36]. They are commonly used as a fundamental vocabulary for domain-specific ontologies to extend. Compared with the global schema in traditional information integration, upper ontologies need to be more general so that they could encompass the top level for the domain-specific ontologies yet to be developed [37]. Some well-known upper ontologies are OpenCyc[1], WordNet[2], Suggested Upper Merged Ontology (SUMO)[3], Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)[4] and Basic Formal Ontology

[1] http://www.opencyc.org/
[2] http://wordnet.princeton.edu/
[3] http://www.ontologyportal.org/
[4] http://www.loa-cnr.it/DOLCE.html

(BFO)[5]. OpenCyc is the subset and open source version of the Cyc knowledge base, which is a common sense ontology containing a large amount of fundamental human knowledge [38]. Though Cyc knowledge base is implemented in CycL language, OpenCyc has been translated and published as an OWL file, used in a wide range of applications, such as multi-agent systems [39], word sense disambiguation [40], question answering [41]. As a freely available lexical database for English words, WordNet is a useful tool in the areas of linguistics, information retrieval, word sense disambiguation, semantic concordance building, text analysis, and knowledge engineering. It groups nouns, verbs, adjectives and adverbs into sets of synonyms called synsets, which are further interrelated via conceptual-semantic and lexical relations [42]. For a better use of WordNet in the Semantic Web community, W3C is currently converting WordNet into RDF/OWL representation language [43]. SUMO is a large standard ontology defined by the IEEE Standard Upper Ontology Working Group with purpose for building a semantic foundation for promoting data interoperability, information retrieval, automated inference, and natural language processing [44, 45]. It is created by merging a number of existing upper-level ontologies and organizing them into a single, comprehensive, and cohesive structure. Till now, SUMO has been mapped to all of the WordNet lexicon and translated into OWL. DOLCE is a formal foundational ontology developed as a part of WonderWeb Foundational Ontologies Library, which is funded by the Information Society Technologies (IST) Programme of the Commission of the European Communities. Described using first-order logic, DOLCE aims at “capturing the ontological categories underlying natural language and human commonsense” [46]. Developed by the Institute for Formal Ontology and Medical Information Science (IFOMIS) at the University of Leipzig, BFO is a formal ontology which focuses on integrating domain ontologies developed for scientific research. It consists of a series of sub-ontologies, the most important of which are a series of snapshot ontologies (called SNAP) and a single videoscopic ontology (called SPAN) [47].

Compared with upper ontologies, domain ontologies serve to model a specific domain, which usually contain domain specific terminologies. Examples of domain ontologies are legal ontologies [48], chemical ontologies [49], spatial ontologies [50], multimedia ontologies [51] and geographical ontologies [52]. One of the domains which widely use the Semantic Web ontologies more than others is the field of biomedicine, where a vast amount of biomedical data on the Web need to be accessed and queried efficiently [53]. The work on ontologies has been recognized as essential in some of the grand challenges of genomics research [7] and many biomedical ontologies have been developed under the international research cooperations, e.g. Medical Subject Headings (MeSH)[6], Systematized Nomenclature of Medicine - Clinical Terms (SNOMED-CT)[7], Gene Ontology (GO)[8] and Open Biological and Biomedical Ontologies (OBO)[9]. Developed by the U.S. Na-

[5] http://www.ifomis.org/bfo/
[6] http://www.nlm.nih.gov/mesh/
[7] http://www.ihtsdo.org/
[8] http://www.geneontology.org/
[9] http://www.obofoundry.org/

tional Library of Medicine (NLM) in 1960, MeSH is a controlled vocabulary in a hierarchical structure. It is primarily used as an index of articles from 5400 biomedical journals, facilitating search queries in a more accurate and efficient way. Some attempts have taken place to translate MeSH into OWL [54]. Being a DL-based biomedical terminology, SNOMED-CT [55] is formed by converging two major clinical terminologies, SNOMED Reference Terminology (SNOMED RT) and Clinical Terms Version 3 [56]. Being a standard clinical language, it facilitates the communication and interoperability in electronic health data exchange. GO is a structured controlled vocabulary with precise and common definitions for describing the roles of genes and gene products within an organism [57]. Currently, it has been a de facto standard with terms widely used for annotating many biological data sources [58]. With more and more researchers starting to generate biomedical ontologies, OBO was established in 2001 as an umbrella body to coordinate the development of biomedical ontologies. To avoid the syntactic and semantic heterogeneity among ontologies, OBO has identified 10 principles for the member ontologies to follow [10]. Nowadays, many bio-ontologies are available via OBO.

Even though the development of upper ontologies makes allowance for the future extension with domain ontologies, it is usually not an easy task to extend upper ontologies with various domain ontologies in a real setting. The extension is hindered by the different constraints and distinct level of conceptualization at their interface. To guarantee a seamless transition between upper ontologies and domain ontologies, a mediating layer – *upper-domain ontologies* (or top-domain ontologies), has been proposed, which serves to define the fundamental concepts and relations relevant to the entire domain. Examples of upper-domain ontologies in the biomedical domain include Ontology of Biomedical Reality (OBR)[10] and Top-Domain Ontology for the Life Science (BioTop)[11]. Intended for facilitating inference across the boundaries of domain ontologies in anatomy, physiology and pathology, OBR is developed for the integration of the upper ontology BFO and three specific domain ontologies. Inheriting the ontological structure of BFO, it harmonizes the design principles of upper ontologies and domain ontologies [59]. BioTop is an upper-domain ontology initially created to redesign and expand GENIA ontology[12], an ontology applied for corpus annotation in molecular biology. With continuous expansion, BioTop currently provides upper-level categories to ontologies in the field of molecular biology, bio-medicine and bio-chemistry [60]. It exhibits links to the upper ontologies DOLCE, BFO and the OBO relation ontology [61], and provides mappings to OBO ontologies [62].

2.2 Ontology Alignment

Many ontologies have been developed in recent years, and many of them contain overlapping information. For instance, OBO lists 38 different anatomy ontologies

[10]http://ontology.buffalo.edu/medo/biomedo.htm
[11]http://www.imbi.uni-freiburg.de/ontology/biotop
[12]http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA

(December 2010). Often we would want to be able to use multiple ontologies. For instance, companies may want to use community standard ontologies and use them together with company-specific ontologies. Applications may need to use ontologies from different areas or from different views on one area. Ontology designers may want to use already existing ontologies as the basis for the creation of new ontologies by extending the existing ontologies or by combining knowledge from different smaller ontologies. In each of these cases, it is important to know the relationships between the terms in the different ontologies. Further, the data in different data sources in the same domain may have been annotated with different but similar ontologies. Knowledge of the inter-ontology relationships would in this case lead to improvements in search, integration and analysis of data. It has been realized that this is a major issue and much research has recently been done on ontology alignment, i.e. finding mappings between terms in different ontologies [63].

2.2.1 Ontology Alignment Framework

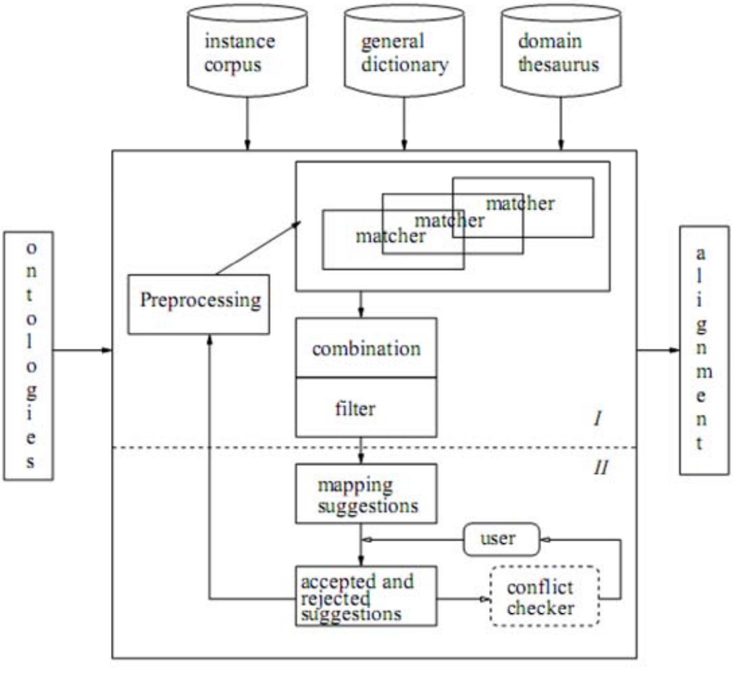

Figure 2.1: Alignment framework [10] (Extension of the framework in [13]).

A large number of ontology alignment systems have been developed (see, for instance, review papers [13, 64, 65, 66], the book [67] on ontology matching, and the ontology matching web site at http://www.ontologymatching.org/). Many ontology alignment systems are based on the computation of similarity values between entities in different ontologies and can be described as instantiations of the general framework in figure 2.1 [10]. The framework consists of two parts. The first part (*I* in figure 2.1) computes mapping suggestions. The second part (*II*) interacts with the user to decide on the final alignment.

An alignment algorithm receives as input two source ontologies. A preprocessing step can be used to modify the original ontologies or to partition the ontologies into mappable parts. The algorithm can include several matchers that calculate similarities between the entities from the different source ontologies or mappable parts of the ontologies. They can implement strategies based on linguistic matching, structure-based strategies, constraint-based approaches, instance-based strategies, strategies that use auxiliary information or a combination of these. Each matcher utilizes knowledge from one or multiple sources. Mapping suggestions are then determined by combining and filtering the results generated by one or more matchers. By using different matchers and combining and filtering the results in different ways we obtain different alignment strategies. The suggestions are then presented to the user who accepts or rejects them. The acceptance and rejection of suggestions may influence further suggestions. Further, a conflict checker is used to avoid conflicts introduced by the mapping suggestions. The output of the alignment algorithm is an alignment which is a set of mappings between entities from the source ontologies.

2.2.2 Ontology Matching Strategies

As the crucial part of an ontology alignment system, matchers measure the similarities between entities from the two source ontologies to be aligned, whose results determine the mappings suggestions generated afterwards. The design of matchers has to take the characteristics of ontologies into account, which may vary with the related applications and domains. Therefore, current ontology alignment systems usually exploit multiple matchers with different matching strategies, which can be categorized from different perspectives [68] [69]. Here, we adopt the categorization in [13], classifying matching strategies based on the resources they use for measuring similarities.

- **Strategies based on linguistic matching**. Most ontologies have their concepts and relations defined using some textual descriptions, such as the names, the labels, the comments and synonyms, which are used as primary resources by most ontology alignment systems for calculating the similarities. Frequently used linguistic-based matching approaches are string-based matching algorithms combined with natural language processing techniques.

- **Structure-based strategies**. For ontologies with structural relations (e.g. *is-a* and *part-of* relations), the similarities of concepts are also implied in the context of the hierarchy. For example, if two concepts from two ontologies are similar, it is likely that their parents (or ancestors) and their children (or descendants) are also similar. Taking advantage of such characteristics, structure-based strategies usually employ some heuristics, such as propagating the similarity via relations between concepts.

- **Constraint-based approaches**. Constraint-based approaches utilize the keys and relationships of the entities in ontologies, such as the datatype, range and domain of their properties. For instance, relations with the same range and domain are likely to be similar, as well as the concepts related by them.

- **Instance-based strategies**. Some ontologies have instances defined in relation to the concepts, which provides an opportunity to measure the similarity of concepts based on instances. Besides these instances available in ontologies, some strategies create and utilize instances from some external resources. For example, the strategy in [70] uses life science literature, in which the similarity between concepts are computed based on the probability that documents about one concept are also about the other concept.

- **Use of auxiliary information**. Auxiliary information is used by many ontology alignment systems. They are usually domain knowledge represented by standard dictionaries and widely recognized thesauri. For systems with an iterative mapping process, previous matching decisions, like user's acceptance/rejection on mapping suggestions, are deemed as an auxiliary information to use.

- **Combining different approaches**. All the above strategies are deemed as local methods in [68] for determining the similarities between entities from different ontologies. In order to generate better mapping suggestions, most ontology alignment systems combine these strategies or their results in the real settings. It is usually accompanied with different filtering strategies to refine the compound results.

2.2.3 Ontology Alignment Evaluation Initiative

With a lot of research efforts put into the area of ontology alignment, more than 30 ontology alignment systems have been developed in recent years [10], proposing matching strategies from different perspectives. To compare, analyze and improve different techniques, researchers realized that a consensus should be established for the evaluation of these methods. For this purpose, the first competitive evaluation of ontology alignment systems was held by Information Interpretation and Integration Conference (I3CON) in 2004, where test cases were setup based on 8 pairs of real-life ontologies. The contest required participants to run one tool in one configuration on all the tests, and the evaluation focused on the quality of the ontology alignment results. Also in 2004, another contest was held by the 3rd Evaluation of Ontology-based Tools (EON) Workshop. Specifically, the contest aims to test the performance of alignment systems on the ontologies with particular features and participants are given the reference alignment to tune their tools. 5 and 4 teams joined the contest of I3CON and EON, respectively.

Ever since 2005, the evaluation has been held within Ontology Alignment Evaluation Initiative (OAEI, http://oaei.ontologymatching.org/) as a yearly initiative. Aiming to assess the strengths and weaknesses of alignment systems, to compare different techniques and to improve evaluation techniques, OAEI has the collective goals from the previous two contests. The ontology alignment problems published by OAEI contain different cases, some of which are open (reference alignment is known beforehand), but most are blind (reference alignment is not known - participants send their mapping suggestions to organizers who evaluate the performance). OAEI currently only evaluates the non-interactive part of ontology alignment systems. The quality of ontology alignment results are measured based on *precision*, *recall*, and *f-measure*. *Precision* measures how many of the mapping suggestions are correct. It is defined as the number of correct suggestions divided by the number of suggestions. *Recall* measures how many of the correct mappings are found by the alignment algorithm. It is defined as the number of correct suggestions divided by the number of correct mappings. *F-measure* is the weighted harmonic mean of precision and recall. In 2010, the competition involves 15 participants worldwide.

2.3 Ontology Debugging

Benefiting from the progress of ontology learning techniques [71], the development of ontologies has been facilitated in recent years. Many ontologies are generated containing thousands of terms with complex relations. Meanwhile, as mentioned before, to be empowered with the automatic reasoning, more and more ontologies are migrating to the logic-based formalism, using DL-based ontology language (e.g. OWL). These efforts improve the status of ontologies, but also bring more challenges. As ontologies become larger and more complex with diverse logic constraints, they are suspectable to all kinds of defects. Some of these defects are hard to diagnose and fix even for domain experts. Ontology debugging approaches are needed to detect and repair the defects in the ontologies.

2.3.1 Ontology Defects

Defects in ontologies can take different forms (e.g. [72]). *Syntactic defects*, including incorrect format or invalid statements, are usually found within not well-formed XML ontology documents. Syntactic defects could also happen to the ontologies, which are composed in a specific expressive sub-language (e.g. OWL-Lite, OWL-DL and OWL-Full), but misuse constructors or primitives outside the desired language level.

Being another kind of defects, *style defects* usually include such things as unintended redundancy and unused concepts or relations. Even though they are not syntactically or semantically invalid, style defects are usually deemed as unintended results from the modeling point of view, which ontology designers should be aware of.

More interesting and severe defects are the *modeling defects*, which refer to the flawed or incongruous modeling of the intended domain, instantiated as wrong or incomplete concepts or relations in ontologies. They appear in ontologies for several reasons. Firstly, the development of ontologies is often restricted by the knowledge and perspective of ontology designers. It is possible that, the designer, even as a domain expert, could have misunderstood or incomplete knowledge about his or her area, which affects the correctness and completeness of the resulted ontologies. On the other hand, the development of ontologies requires the techniques of knowledge representation. It is likely that people who are not expert in knowledge representation could misuse and confuse the primitives in the ontology language (e.g. is-a and part-of relations), which leads to modeling defects in the resulted ontologies.

With more and more ontologies starting to use the DL-based ontology language, *semantic defects* become another common and important kind of ontological defects. Semantic defects take the form of logical contradictions, which could be categorized into two cases: *inconsistency* and *incoherency*. An inconsistent ontology is defined as one which has no model. An incoherent ontology is defined as one which has an unsatisfiable concept, while an *unsatisfiable concept* is a concept which cannot have any instance (i.e. represents the empty set) [73]. Semantic defects could be introduced by the modeling defects, the integration of ontologies, migration from other formalisms and ontology evolution [74].

2.3.2 Debugging Defects in Ontologies

Generally, syntactic defects are the easiest to fix. Using XML parsers (e.g. Xerces[13]) or RDF validators[14], the ontology designers could easily locate the lines or characters which make the document syntactically invalid and fix them. Regarding the improper use of primitives beyond an intended language level, there are also tools, such as OWL Species Validation tools[15], helping the user to identify the defects and bring the ontologies to the intended language level.

Style defects in ontologies are also easy to detect. For instance, using the Racer [75] reasoner, the ontology checker in SAMBO [13] could easily find cycles in the definitions of concepts and redundant subsumption relations. However, whether and how to deal with the style defects depends on the intent of the ontology designer. Currently, the debugging of style defects is often done by exposing the problems to the user.

Being the incorrect or incomplete conceptualization of the intended domain, the detecting of modeling defects usually requires the domain knowledge, which may come from domain experts, or external resources such as text corpus, dictionary or knowledge base. For ontologies using the logic-based language, some inherent modeling defects, such as wrongly-defined concepts, may cause logical contradictions and thus be found by the logical reasoners. Another possible way of

[13]http://xerces.apache.org/xerces-j/
[14]http://www.w3.org/RDF/Validator/
[15]http://phoebus.cs.man.ac.uk:9999/OWL/Validator

finding modeling defects is testing the ontologies in the intended applications and tracing the defects in case of malfunctions of applications. Regarding the repairing of modeling defects, it is usually a semi-automatic process, which requires domain knowledge and interactions with the ontology designer.

The debugging of semantic defects aims at identifying and removing logical contradictions, i.e. inconsistencies and incoherencies, from an ontology. They use standard reasoners, such as Pellet [76], FaCT [77] and Racer, to identify the existence of a contradiction, and provide support for resolving and eliminating it [78]. It is usually done by identifying the axioms responsible for the logical contradictions, and generating suggestions to delete some axioms or weaken them. During this process, the minimal change (i.e. the minimal loss of information) to the ontology, is often preferred [74].

Chapter 3

Aligning Ontologies using Partial Reference Alignments

In this chapter, we deal with the missing mappings between networked ontologies. Given two ontologies connected via a set of correct mappings (PRA), we use these mappings to find more correct mappings between these two ontologies. Several strategies of using the PRA are proposed regarding the different components of a general ontology alignment framework. We conduct experiments based on the alignment systems SAMBO and SAMBOdtf and discuss the results.

3.1 Introduction

In the current ontology alignment systems, an alignment is computed from scratch. However, recently, some situations have occurred where a PRA is available, i.e. some of the correct mappings between terms are given or have been obtained. One example is the development of Bioportal [79] where mappings between different ontologies in the biomedical domain have been collected (although not all mappings are validated). Bioportal also supports collaborative ontology alignment (one of the challenges for ontology alignment described in [9]) where experts can focus on their piece of expertise. In this case for some parts of the ontologies mappings will be available while they are still lacking for other parts of the ontologies. Another situation is an iterative ontology alignment methodology where people and ontology alignment systems interact to iteratively align and improve the quality of the mappings. In systems such as SAMBO [13], users can input their own mappings as well as accept or reject mapping suggestions generated by the system. Both the mappings given by the users and the rejection and acceptance of system-generated mappings by users influence future iterations of ontology alignment. Finally, the Ontology Alignment Evaluation Initiative (OAEI, http://oaei.ontologymatching.org/) introduced in 2008 a new task in the Anatomy track [80] where a PRA was given and participating systems could use the PRA to improve the quality of their mapping suggestions.

In this chapter, we investigate whether and how a PRA can be used in ontology alignment. We use PRAs in the different steps of ontology alignment. We use PRAs in a preprocessing step to partition the ontologies into mappable parts that are likely to contain correct mappings and therefore not every term in the first ontology needs to be compared to every term in the second ontology. We also use PRAs in the computation of similarities between terms. Further, we use PRAs to filter mapping suggestions. We test the approaches on previously developed gold standards (from [13] and the Anatomy track of OAEI) and discuss the results.

The remainder of the chapter is organized as follows. In section 3.2 we describe the systems, the gold standards and the evaluation measures that we have used in our experiments. The proposed approaches and experiments are described in section 3.3. We investigate the use of PRAs in the different components of the ontology alignment framework. We also investigate the influence of the size of the PRA on the quality of the alignments. The results of the experiments are shown and discussed in section 3.4. Related work is presented in section 3.5 and the chapter is summarized in section 3.6.

3.2 Preliminaries

3.2.1 SAMBO and SAMBOdtf

SAMBO and SAMBOdtf are ontology alignment systems based on the framework described in Figure 2.1 of the previous chapter. They both currently contain five basic matchers [13]: two terminological matchers (a basic matcher and an extension using WordNet; extension described below), a structure-based matcher (which uses the is-a and part-of hierarchies of the source ontologies), a matcher based on domain knowledge (described below), and a learning matcher (which uses literature that is related to the concepts in the ontologies to define a similarity value between the concepts). In addition to these techniques also other matchers were developed [81, 70].

In our evaluations we use the versions of SAMBO and SAMBOdtf as used in OAEI 2008 [63]. These systems performed best and second best, respectively, in the Anatomy track of OAEI 2008. These versions contain the matchers *TermWN* and *UMLSKSearch*. The matcher *TermWN* contains matching algorithms based on the textual descriptions (names and synonyms) of concepts and relations. In the current implementation, the matcher includes two approximate string matching algorithms (n-gram and edit distance), and a linguistic algorithm that also uses WordNet to find synonyms and is-a relations. The aggregated similarity value generated from TermWN has range [0,1]. Our matcher *UMLSKSearch* uses the Metathesaurus in the Unified Medical Language System (UMLS)[1]. The similarity of two terms in the source ontologies is determined by their relationship in UMLS. In our experiments we used the UMLS Knowledge Source Server to query the UMLS

[1] http://www.nlm.nih.gov/research/umls/

Metathesaurus with source ontology terms. The querying is based on searching the normalized string index and normalized word index provided by the UMLS Knowledge Source Server. We used version 2008AA of UMLS. As a result we obtain concepts that have the source ontology term as their synonym. We assign a similarity value of 0.99 if the source ontology terms are synonyms of the same concept and 0 otherwise.

The combination algorithm used for the OAEI versions of SAMBO and SAMBOdtf is a maximum-based algorithm. The similarity value for a pair of terms is the maximum of the values obtained from TermWN and UMLSKSearch for this pair of terms.

The filtering method in SAMBO is single threshold filtering. Pairs of concepts with a similarity value higher than or equal to a given threshold value are returned as mapping suggestions to the user. SAMBOdtf implements the double threshold filtering method developed in [82]. The double threshold filtering approach uses the structure of the ontologies. It is based on the observation that (for the different approaches in the evaluation in [13]) for single threshold filtering the precision of the results is decreasing and the recall is increasing when the threshold is decreasing. Therefore, we propose to use two thresholds. Pairs with similarity value equal or higher than the upper threshold are retained as suggestions. The intuition is that this gives suggestions with a high precision. Further, pairs with similarity values between the lower and the upper threshold are filtered using structural information and the rest is discarded. We require that the pairs with similarity values between the two thresholds are 'reasonable' from a structural point of view.[2] The intuition here is that the recall is augmented by adding new suggestions, while at the same time the precision stays high because only structurally reasonable suggestions are added. The double threshold filtering approach contains the following three steps. (i) Find a consistent suggestion group from the pairs with similarity value higher or equal than the upper threshold. We say that a set of suggestions is a consistent suggestion group if each concept occurs at most once as first argument in a pair, at most once as second argument in a pair and for each pair of suggestions <A, A’> and <B, B’> where A and B are concepts in the first ontology and A’ and B’ are concepts in the second ontology: A ⊂ B iff A’ ⊂ B’. (ii) Use the consistent suggestion group to partition the original ontologies. (iii) Filter the pairs with similarity values between the lower and upper thresholds using the partitions. Only pairs of which the elements belong to corresponding pieces in the partitions are retained as suggestions. For details we refer to [82].

In contrast to the original versions of SAMBO and SAMBOdtf where a term in one ontology can be suggested to be mapped to different terms in the other ontology, in the OAEI versions this is not the case. We retain only suggestions where the similarity between the terms in the suggestion is higher than or equal to the similarity of these terms to any other term according to the suggestion list. In the case there are different possibilities, one is randomly chosen. (In the implementation the first in the list is chosen.)

[2]In our implementation we have focused on the is-a relation.

3.2.2 Gold Standard Alignments

In our tests we use alignments that have been used in previous evaluations ([13] and OAEI). Essentially, for our purpose all used ontologies can be seen as taxonomies (concepts and is-a and part-of relations). In [13] a number of smaller test cases were introduced to evaluate different alignment strategies. For the first two cases we use a part of a Gene Ontology (GO) [57] together with a part of Signal Ontology (SigO) [83]. The first case, *B* (behavior), contains 57 terms from GO and 10 terms from SigO. Its reference alignment (RA) contains 4 mappings. The second case, *ID* (immune defense), contains 73 terms from GO and 17 terms from SigO. Its RA contains 8 mappings. The other cases are taken from the anatomy category of Medical Subject Headings (MeSH)[3] and the Adult Mouse Anatomy (MA, available from OBO): *nose* (15 terms from MeSH, 18 terms from MA, and 7 mappings in the RA), *ear* (39 terms from MeSH, 77 terms from MA, and 27 mappings in the RA), and *eye* (45 terms from MeSH, 112 terms from MA, and 27 mappings in the RA).

A larger test case that we use is the case of the Anatomy track of OAEI (version 2008), where participants were required to align the Adult Mouse Anatomy (2744 concepts) and the NCI Thesaurus - anatomy (3304 concepts). The RA contains 1523 equivalence mappings of which 934 are deemed trivial (i.e. they can be found by a relatively basic string-based matcher).

3.2.3 Evaluation Measures

The results of our experiments are given in terms of the quality of the mapping suggestions. We use *precision*, *recall*, $recall_{PRA}$ and *f-measure*. As mentioned in Chapter 2, *Precision* measures how many of the mapping suggestions are correct. It is defined as the number of correct suggestions divided by the number of suggestions. *Recall* measures how many of the correct mappings are found by the alignment algorithm. It is defined as the number of correct suggestions divided by the number of correct mappings. We also introduce the measure $recall_{PRA}$ which measures how many of the correct mappings that are not in a PRA are found by the alignment algorithm. It is defined as the number of correct mapping suggestions not in the PRA divided by the number of correct mappings not in the PRA. *F-measure* is the weighted harmonic mean of precision and recall. In our tests precision and recall are weighted evenly.

3.3 Proposed Approaches and Experiments

In this section we define a number of experiments to test the usefulness of using a PRA for ontology alignment. We consider the use of PRAs in the different components of the alignment framework in figure 2.1. As base systems we use SAMBO and SAMBOdtf as described in section 3.2.1 and modify their components. We de-

[3] http://www.nlm.nih.gov/mesh/

	preprocessing	matchers	combination	filter
SAMBO	none	TermWN + UMLSKSearch	maximum	single threshold
SAMBOdtf	none	TermWN + UMLSKSearch	maximum	double threshold
mgPRA	partitioning	TermWN + UMLSKSearch	maximum	single threshold filter with PRA
mgfPRA	fixing and partitioning	TermWN + UMLSKSearch	maximum	single threshold filter with PRA
pmPRA	none	TermWN + UMLSKSearch pattern-based augmentation	maximum	single threshold filter with PRA
fPRA	none	TermWN + UMLSKSearch	maximum	single threshold filter with PRA
dtfPRA	none	TermWN + UMLSKSearch	maximum	double threshold with PRA filter with PRA
pfPRA	none	TermWN + UMLSKSearch	maximum	filter based on EM and PRA filter with PRA

Table 3.1: Alignment strategies.

scribe the experiments and provide an overview of the resulting strategies in table 3.1.

An immediate observation regarding a PRA is that the mappings in the PRA are deemed to be correct and therefore should be included in the final result. For the same reason, if there are different suggestions for mapping a term, then a suggestion contained in the PRA is preferred. Therefore, in all alignment strategies using a PRA, we add the mappings in the PRA to the list of suggestions with a special status. These mappings cannot be removed in any filtering step. This technique we call *filter with PRA*.

3.3.1 Use of PRA in the Preprocessing Step

A first question we want to investigate is whether we can use a PRA in the preprocessing phase. Most systems compute a similarity value between all terms from the first ontology and all terms from the second ontology. Some work has been done on partitioning the ontologies to find mappable parts of the ontologies (e.g. [84]). The motivation for that work is scalability. When the size of the ontologies grows, the performance of most current ontology matching techniques becomes worse. In our work we investigate whether we can use a PRA to partition the ontologies into mappable parts and test whether, in addition to the fact that we do not have to compute similarity values between all terms from the first ontology and all terms from the second ontology, this also leads to a better quality of the mapping suggestions.

In the first approach we partition the ontologies into mappable parts using the partitioning step of the double threshold filtering described in section 3.2.1 and [82]. A part of the PRA satisfying the consistent group property is used as a consistent group. The resulting alignment strategy (this preprocessing, the SAMBO matchers, combination and filters, and filter with PRA) we call *mgPRA* (mappable groups with PRA).

According to our experience in aligning ontologies we know that the structure of the source ontologies is not always perfect. For instance, given the two ontologies and the PRA in the Anatomy task of OAEI 2008, it can be deduced that many is-a relations are missing in at least one of the source ontologies. Based on this observation we experiment with a second approach where we add to the source ontologies the missing is-a relationships that can be deduced from the source ontologies and the PRA.[4] After this 'fixing' of the source ontologies the PRA will satisfy the consistent group property. The resulting strategy is called *mgfPRA* (mappable groups and fixing with PRA).

3.3.2 Use of PRA in a Matcher

One way to create a matcher based on a PRA, is to use underlying properties of the mappings in the PRA. We have previously observed that sometimes for two given source ontologies, common patterns can be found between the correct mappings. For instance, in the PRA of the OAEI 2008 Anatomy we find the mappings <lumbar vertebra 5, l5 vertebra> and <thoracic vertebra 11, t11 vertebra> which both have the second term made up of three elements from the first term (the concatenation of the first character in the first token and the third token, then a blank and the second token). Also the mappings <forebrain, fore brain>, <gallbladder, gall bladder> (in which the first term is the concatenation of the only two tokens from the second term), and the mappings <stomach body, body stomach> and <stomach fundus, fundus stomach> (in which the first term is made up from the tokens from the second term in reverse order) share similar linguistic patterns, respectively. When using different linguistic matchers the similarity values according to these matchers for the mappings sharing similar patterns are therefore very similar as well.

Based on this observation we developed a matcher that augments previously generated similarity values for term pairs when these term pairs display a similar (linguistic) pattern as mappings in the PRA. For the experiment we used TermWN and UMLSKSearch to compute the original similarity values. Further, for each term pair we compute a vector with, in this case, three similarity values, based on the different components of TermWN (n-gram, edit distance and a linguistic algorithm that uses WordNet; as described in section 3.2.1). Based on this similarity vector we compute the Euclidean distance of a term pair to the mappings in the PRA and count how many of the PRA mappings are within a predefined radius. (In the experiment we used 0.1 for the radius.) Based on the value of this count, the matcher may augment the original similarity value for the term pair. (In the experiment we used (count $\times$ 0.06) as augmentation value and augmented only pairs with a similarity value lower than 0.9. We also set the limit for the augmented similarity value to 0.9.) The resulting alignment strategy is called *pmPRA* (pattern matcher with PRA).

[4]This is in line with the challenge for ontology alignment on discovering missing background knowledge as described in [9].

3.3.3 Use of PRA in the Filter Step

Another question that we want to investigate is whether we can use a PRA for filtering the list of mapping suggestions.

As mentioned before, all strategies using PRAs implement the filter with PRA approach. In a first experiment we only add this filter approach to SAMBO. The resulting alignment strategy is called *fPRA*.

In addition to filter with PRA, the second strategy also uses a variant of the double threshold filtering approach of SAMBOdtf. While in the original double threshold filtering approach, a consistent suggestion group is computed based on mapping suggestions with a high similarity value, in this approach a part of the PRA satisfying the consistent group property is used as a consistent group. The resulting alignment strategy is called *dtfPRA*.

The last strategy we use is called *pfPRA* (pattern filter with PRA). This strategy is based on the observation described above that some correct mappings share similar patterns. Similar to pmPRA, we assign a vector with similarity values based on the different components of TermWN to each term pair in the suggestion list and the PRA. We compute clusters based on these similarity vectors of the term pairs in the suggestion list using the expectation-maximization algorithm. (For the small cases we set the number of clusters. For *Anatomy* we did not use a predefined number of clusters.) Then, each mapping in the PRA is assigned to the cluster for which the distance between the cluster center and the PRA mapping is the smallest. Further, we compute for each cluster the average distance between a PRA mapping in the cluster and its cluster center. The filter strategy retains the suggestions in a cluster with a distance to the cluster center that is smaller or equal to the computed average for that cluster and discards the others.

3.3.4 Influence of the Size of the PRA

We also want to investigate the influence of the size of the PRA. For this purpose we compare the results of the approaches for the *Anatomy* case with two PRAs: PRA-F and its subset PRA-H. For PRA-F we take the PRA as provided by the OAEI 2008 Anatomy task. PRA-H contains half of the trivial and half of the non-trivial mappings from PRA-F.

3.4 Results and discussion

Table 3.2 shows the number of suggestions generated by the algorithms in the experiments. For SAMBO and SAMBOdtf we also added the number of mappings in the PRA that were found by those algorithms. We note that for all thresholds for *B*, *nose* and *ear*, for thresholds 0.4 and 0.6 for *ID* and for threshold 0.4 for *eye* SAMBO actually finds all mappings in the PRA. This means that in these cases adding the PRA to the solutions in itself does not improve recall. For *Anatomy* SAMBO does not find between 45 (threshold 0.4) and 55 (threshold 0.8) of the 988 mappings in the PRA.

Case	*Th*	SAMBO	SAMBOdtf	mgPRA	mgfPRA	pmPRA	fPRA	dtfPRA	pfPRA
B	0.4	6/2	6/2	6	2	6	6	4	3
	0.6	5/2	5/2	5	2	5	5	4	3
	0.8	4/2	-	4	2	4	4	-	3
ID	0.4	12/4	11/3	12	12	12	12	11	12
	0.6	8/4	7/3	5	5	8	8	8	8
	0.8	7/3	-	4	5	8	8	-	8
nose	0.4	7/4	7/4	7	4	7	7	7	6
	0.6	7/4	7/4	7	4	7	7	7	6
	0.8	7/4	-	7	4	7	7	-	6
ear	0.4	30/14	29/14	28	18	4	30	30	25
	0.6	29/14	29/14	27	18	3	29	29	25
	0.8	26/14	-	24	18	1	26	-	24
eye	0.4	31/13	30/13	30	13	6	31	31	23
	0.6	36/12	26/12	26	13	3	27	27	22
	0.8	24/11	-	25	13	2	26	-	22
Anatomy	0.4	1575/943	1527/940	1690	1663	1625	1601	1552	1251
	0.6	1466/942	1438/940	1488	1444	1598	1498	1474	1221
	0.8	1297/933	-	1308	1271	1528	1342	-	1139

Table 3.2: Number of mapping suggestions. For SAMBO and SAMBOdtf also the number of PRA mappings found.

The results in terms of precision, recall, f-measure and recall$_{PRA}$ are given in tables[5] 3.3 to 3.7. The results in the tables are truncated values. Note also that as SAMBO and SAMBOdtf do not use a PRA, we have set recall = recall$_{PRA}$ for SAMBO and SAMBOdtf. *Th* is the threshold for the filtering for the single threshold approaches. For SAMBOdtf and dtfPRA the upper threshold is always 0.8 while *Th* is the lower threshold. There are no results for SAMBOdtf and dtfPRA for upper threshold 0.8 as this would be the same as using SAMBO and fPRA with single threshold 0.8, respectively.

3.4.1 PRA in Preprocessing

The results of the experiments for using PRAs in the preprocessing step are given in table 3.3. The intuition behind mgPRA and mgfPRA is to partition the ontologies into mappable parts. Therefore, we can only generate mapping suggestions that are reasonable from a structural point of view. This suggests that, comparing to the base systems, the precision may become higher as suggestions that do not conform to the structure of the source ontologies cannot be made. As we add the PRA to the result, the recall may be increased as some of the PRA mappings may not be found by the base systems. However, the similarity values between the terms do not change and it is therefore not likely that new mappings are found. (The only way to find new mappings compared to the base system is when a mapping suggestion with high similarity in the base system cannot be suggested by using mgPRA and mgfPRA because the terms were in incompatible parts of the ontologies. In that

[5] As *Anatomy* is a blind case at OAEI we do not have the RA available. Therefore, we have sent the mapping suggestions for *Anatomy* to the organizers of the OAEI 2008 Anatomy track who have returned the values for the different evaluation measures.

Case	RA	PRA	*Th*	SAMBO	mgPRA	mgfPRA
B	4	2	0.4	0.66/1.00/0.80/1.00	0.66/1.00/0.80/1.00	1.00/0.50/0.66/0.00
			0.6	0.80/1.00/0.88/1.00	0.80/1.00/0.88/1.00	1.00/0.50/0.66/0.00
			0.8	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00	1.00/0.50/0.66/0.00
ID	8	4	0.4	0.50/0.75/0.60/0.75	0.41/0.62/0.50/0.25	0.41/0.62/0.50/0.25
			0.6	0.75/0.75/0.75/0.75	1.00/0.62/0.76/0.25	1.00/0.62/0.76/0.25
			0.8	0.71/0.62/0.66/0.62	1.00/0.62/0.76/0.25	1.00/0.62/0.76/0.25
nose	7	4	0.4	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00	1.00/0.57/0.72/0.00
			0.6	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00	1.00/0.57/0.72/0.00
			0.8	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00	1.00/0.57/0.72/0.00
ear	27	14	0.4	0.86/0.96/0.91/0.96	0.85/0.88/0.87/0.76	1.00/0.66/0.80/0.30
			0.6	0.89/0.96/0.92/0.96	0.88/0.88/0.88/0.76	1.00/0.66/0.80/0.30
			0.8	0.96/0.92/0.94/0.92	1.00/0.88/0.94/0.76	1.00/0.66/0.80/0.30
eye	27	13	0.4	0.80/0.92/0.86/0.92	0.80/0.88/0.84/0.78	1.00/0.48/0.65/0.00
			0.6	0.92/0.88/0.90/0.88	0.92/0.88/0.90/0.78	1.00/0.48/0.65/0.00
			0.8	0.91/0.81/0.86/0.81	0.92/0.85/0.88/0.71	1.00/0.48/0.65/0.00
Anatomy	1523	988	0.4	0.82/0.85/0.83/0.85	0.78/0.87/0.82/0.64	0.78/0.85/0.81/0.58
			0.6	0.88/0.84/0.86/0.84	0.88/0.86/0.87/0.61	0.88/0.84/0.86/0.55
			0.8	0.94/0.80/0.87/0.80	0.96/0.82/0.89/0.50	0.96/0.80/0.88/0.45

Table 3.3: Using the PRA in the preprocessing phase (precision/recall/f-measure/recall$_{PRA}$).

case other suggestions involving these terms may be generated.) The results give some support to these intuitions. For threshold 0.8 the precision of mgPRA and mgfPRA is always equal to or higher than the precision for SAMBO. This is also almost always the case for threshold 0.6. For threshold 0.4 there is no conclusive result. We also notice that, except for threshold 0.4 in *ID* and *Anatomy*, mgfPRA always gives better precision than SAMBO.

As expected, the recall for mgPRA and mgfPRA is equal to or less than the recall for SAMBO in most cases. For the large test case the recall is always higher for mgPRA and equal for mgfPRA. In the cases there is a loss of recall, this is due to a different modeling in the source ontologies. When the PRA satisfies the consistent group property, mgPRA and mgfPRA give the same results. When this is not the case, mgfPRA 'fixes' the source ontologies by adding missing is-a relationships. In most cases this has led to an improvement in precision, in the other cases the precision stayed the same. However, contrary to the intuition in many cases fixing the source ontologies has led to a decrease in recall. This is due to the use of is-a in the source ontologies. For instance, the hierarchical relation in MeSH covers both is-a and part-of. Therefore, not all of these relations should be treated as is-a. However, as mgfPRA cannot distinguish between these, it may fix the source ontology in the wrong way, by adding a hierarchical link (which mgfPRA interprets as is-a, but which should have been interpreted as part-of). For instance, in the *nose* case, having <nose, nose> in the PRA would lead to introducing is-a relations in MA between nose and its parts. Therefore, fixing the ontologies may lead to worse results. As for all alignment strategies using structural information, the quality of the underlying ontologies, the completeness of the structure and the correct use of the structural relations, has an important influence on the quality of the results.

3.4.2 PRA in a Matcher

The results of the experiment using PRAs in a matcher are found in table 3.4. The intuition of the matcher was to augment the similarity values of suggestions that had a similar linguistic pattern as mappings in the PRA. For *B*, *nose* and *ear* the results for SAMBO and pmPRA are the same. The augmentation did not have any influence. For *B* and *nose* the recall was already 1, so that no influence was expected. For *ear* the found correct mappings had already a high similarity value and the missed correct mappings (two for threshold 0.8 and one for thresholds 0.4 and 0.6) did not have a similar linguistic pattern as the mappings in the PRA. For *ID* the recall and the precision were equal or became higher (because of the addition of the PRA to the results). For *eye* the recall improved or was the same. For *Anatomy* the precision and, for low thresholds, also the recall decreased. The recall increased for high thresholds. To investigate the increase of recall for *Anatomy*, we compared the results of using PRA-F with PRA-H. In this case augmenting allowed to find <lateral cuneiform, external cuneiform bone foot> and <brain arachnoid matter, cerebral arachnoid membrane>. For each of these there were 7 mappings in PRA-H with similar linguistic patterns.

Case	RA	PRA	*Th*	SAMBO	pmPRA
B	4	2	0.4	0.66/1.00/0.80/1.00	0.66/1.00/0.80/1.00
			0.6	0.80/1.00/0.88/1.00	0.80/1.00/0.88/1.00
			0.8	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00
ID	8	4	0.4	0.50/0.75/0.60/0.75	0.50/0.75/0.60/0.50
			0.6	0.75/0.75/0.75/0.75	0.75/0.75/0.75/0.50
			0.8	0.71/0.62/0.66/0.62	0.75/0.75/0.75/0.50
nose	7	4	0.4	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00
			0.6	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00
			0.8	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00
ear	27	14	0.4	0.86/0.96/0.91/0.96	0.86/0.96/0.91/0.92
			0.6	0.89/0.96/0.92/0.96	0.89/0.96/0.92/0.92
			0.8	0.96/0.92/0.94/0.92	0.96/0.92/0.94/0.84
eye	27	13	0.4	0.80/0.92/0.86/0.92	0.80/0.92/0.86/0.85
			0.6	0.92/0.88/0.90/0.88	0.89/0.92/0.90/0.85
			0.8	0.91/0.81/0.86/0.81	0.92/0.88/0.90/0.78
Anatomy	1523	988	0.4	0.82/0.85/0.83/0.85	0.78/0.83/0.81/0.54
			0.6	0.88/0.84/0.86/0.84	0.79/0.83/0.81/0.54
			0.8	0.94/0.80/0.87/0.80	0.83/0.83/0.83/0.52

Table 3.4: Using the PRA in a matcher (precision/recall/f-measure/recall$_{PRA}$).

3.4.3 PRA in a Filter

The results for the experiments regarding the use of PRAs in the filter step are given in tables 3.5 and 3.6. In the filter phase mapping suggestions are removed. Both correct and wrong suggestions can be removed and therefore both precision and recall can change compared to the base system. All strategies implement filter with PRA. Therefore, as we add the PRA to the result, the recall may be increased as some of the PRA mappings may not be found by the base systems. The precision and recall may increase when wrong suggestions are removed because there are mappings in the PRA involving the same terms. This intuition is supported by the test cases. For all cases fPRA has a higher or equal precision and recall than SAMBO.

Case	RA	PRA	*Th*	SAMBO	fPRA	pfPRA
B	4	2	0.4	0.66/1.00/0.80/1.00	0.66/1.00/0.80/1.00	1.00/0.75/0.85/0.50
			0.6	0.80/1.00/0.88/1.00	0.80/1.00/0.88/1.00	1.00/0.75/0.85/0.50
			0.8	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00	1.00/0.75/0.85/0.50
ID	8	4	0.4	0.50/0.75/0.60/0.75	0.50/0.75/0.60/0.50	0.50/0.75/0.60/0.50
			0.6	0.75/0.75/0.75/0.75	0.75/0.75/0.75/0.50	0.75/0.75/0.75/0.50
			0.8	0.71/0.62/0.66/0.62	0.75/0.75/0.75/0.50	0.75/0.75/0.75/0.50
nose	7	4	0.4	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00	1.00/0.85/0.92/0.66
			0.6	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00	1.00/0.85/0.92/0.66
			0.8	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00	1.00/0.85/0.92/0.66
ear	27	14	0.4	0.86/0.96/0.91/0.96	0.86/0.96/0.91/0.92	1.00/0.92/0.96/0.84
			0.6	0.89/0.96/0.92/0.96	0.89/0.96/0.92/0.92	1.00/0.92/0.96/0.84
			0.8	0.96/0.92/0.94/0.92	0.96/0.92/0.94/0.84	1.00/0.88/0.94/0.76
eye	27	13	0.4	0.80/0.92/0.86/0.92	0.80/0.92/0.86/0.85	0.95/0.81/0.88/0.64
			0.6	0.92/0.88/0.90/0.88	0.92/0.92/0.92/0.85	1.00/0.81/0.89/0.64
			0.8	0.91/0.81/0.86/0.81	0.92/0.88/0.90/0.78	1.00/0.81/0.89/0.64
Anatomy	1523	988	0.4	0.82/0.85/0.83/0.85	0.83/0.88/0.86/0.66	0.91/0.74/0.82/0.28
			0.6	0.88/0.84/0.86/0.84	0.89/0.87/0.88/0.64	0.93/0.74/0.82/0.27
			0.8	0.94/0.80/0.87/0.80	0.95/0.84/0.89/0.54	0.97/0.72/0.83/0.22

Table 3.5: Using the PRA during the filter phase - 1 (precision/recall/f-measure/recall$_{PRA}$).

In dtfPRA we use, in addition to filter with PRA, also the structure of the source ontologies to filter the suggestion list by filtering out the suggestions which are not reasonable with respect to the structure of the ontologies and the given PRA. The intuition is that wrong suggestions may be removed and that it is better to use a PRA than a computed suggestion group. For *B*, *ID* and *Anatomy*[6] dtfPRA has a higher precision and recall than SAMBOdtf. For *ear* and *eye* the recall is equal or higher for dtfPRA than for SAMBOdtf. For lower threshold 0.4 the precision is lower for dtfPRA than for SAMBOdtf, while it is higher or equal for lower threshold 0.6. The lower precision for *ear* at threshold 0.4 comes from a suggestion

[6] According to the results of OAEI 2008 Anatomy task [80], with respect to the unknown part of the RA, dtfPRA's precision increased with 0.040, its recall with 0.008 and its f-value with 0.025. dtfPRA was the system with the highest increase in f-value and was the only system that used the PRA to increase both precision and recall. In [80] dtfPRA is called 'SAMBOdtf for task 4'.

Case	RA	PRA	*Th*	SAMBOdtf	dtfPRA
B	4	2	0.4	0.66/1.00/0.80/1.00	1.00/1.00/1.00/1.00
			0.6	0.80/1.00/0.88/1.00	1.00/1.00/1.00/1.00
ID	8	4	0.4	0.45/0.62/0.52/0.62	0.54/0.75/0.63/0.50
			0.6	0.71/0.62/0.66/0.62	0.75/0.75/0.75/0.50
nose	7	4	0.4	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00
			0.6	1.00/1.00/1.00/1.00	1.00/1.00/1.00/1.00
ear	27	14	0.4	0.89/0.96/0.92/0.96	0.86/0.96/0.91/0.92
			0.6	0.89/0.96/0.92/0.96	0.89/0.96/0.92/0.92
eye	27	13	0.4	0.83/0.92/0.87/0.92	0.80/0.92/0.86/0.85
			0.6	0.92/0.88/0.90/0.88	0.92/0.92/0.92/0.85
Anatomy	1523	988	0.4	0.84/0.84/0.84/0.84	0.86/0.87/0.87/0.65
			0.6	0.89/0.84/0.86/0.84	0.90/0.87/0.88/0.64

Table 3.6: Using the PRA during the filter phase - 2 (precision/recall/f-measure/recall$_{PRA}$).

<inner ear epithelium, inner hair cell> which was filtered out by SAMBOdtf, but not by dtfPRA. One reason could be that in the SAMBOdtf case the consistent group consisted of 17 mapping suggestions while the consistent part of the PRA only consisted of 9 mappings (as the whole PRA did not satisfy the consistent group property). The partitioning for SAMBOdtf could therefore result in smaller mappable parts. We also compared dtfPRA to SAMBOdtf with just adding the PRA to the results. In most cases we have the same precision and recall. For *ID* threshold 0.4 precision for dtfPRA is slightly better, while for *ear* threshold 0.4 it is slightly worse. For *Anatomy* just adding the PRA to SAMBOdtf actually gives a slightly higher recall than dtfPRA (3 more correct mappings are found), but a lower precision.

The third filter strategy removes suggestions that do not have similar linguistic patterns than the mappings in the PRA. We expect therefore that some correct suggestions obtained through UMLS will be removed and therefore the recall may go down. This is indeed the case in our experiments. The precision for pfPRA is, however, always higher or equal to the precision for SAMBO. This is because the suggestions that had a linguistically similar pattern as mappings in the PRA were usually correct.

3.4.4 Influence of the Size of the PRA

The results for the experiment regarding the influence of the size of the PRA are shown in table 3.7. Intuitively, the more correct mappings we have, the higher the recall should be. This is supported by all test cases. As shown in the last column of table 3.7, several mappings in PRA-F are not found by the algorithms using PRA-H. However, using PRA-H seems to generate more mappings from

Strategy	*Th*	PRA-F	new-F	PRA-H	new-H	NF
mgPRA	0.4	0.78/0.87/0.82	345	0.80/0.85/0.82	351	44
	0.6	0.88/0.86/0.87	327	0.88/0.83/0.85	337	46
	0.8	0.96/0.82/0.89	281	0.95/0.80/0.86	281	50
mgfPRA	0.4	0.78/0.85/0.81	313	0.79/0.81/0.80	336	85
	0.6	0.88/0.84/0.86	295	0.87/0.80/0.83	321	87
	0.8	0.96/0.80/0.88	243	0.95/0.76/0.84	268	89
pmPRA	0.4	0.78/0.83/0.81	290	0.77/0.83/0.80	313	26
	0.6	0.79/0.83/0.81	290	0.79/0.83/0.81	312	26
	0.8	0.83/0.83/0.83	282	0.84/0.82/0.83	294	28
fPRA	0.4	0.83/0.88/0.86	356	0.83/0.86/0.84	357	25
	0.6	0.89/0.87/0.88	347	0.88/0.86/0.87	348	26
	0.8	0.95/0.84/0.89	293	0.95/0.82/0.88	294	30
pfPRA	0.4	0.91/0.74/0.82	152	0.90/0.74/0.81	179	32
	0.6	0.93/0.74/0.82	148	0.92/0.74/0.82	175	33
	0.8	0.97/0.72/0.83	118	0.96/0.71/0.82	136	34
dtfPRA	0.4	0.86/0.87/0.87	350	0.84/0.86/0.85	355	26
	0.6	0.90/0.87/0.88	344	0.89/0.86/0.87	348	26

Table 3.7: *Anatomy* (1523 correct mappings in the RA) with PRA-F (988 mappings) and PRA-H (494 mappings) - (precision/recall/f-measure). new-X represents the number of correct mappings not in PRA-F found by using PRA-X. NF is the number of mappings in PRA-F not found by the algorithms using PRA-H.

the unknown part of the RA. On the other hand, it also generates more wrong suggestions. One explanation could be that for most strategies the larger the PRA, the more constraints the suggestions need to satisfy and thus the fewer suggestions (correct and wrong) are generated. For the preprocessing strategies mgPRA and mgfPRA the precision is lower for the larger PRA when we use a low threshold. However, the precision is better for the larger PRA when the threshold is high. The matcher strategy pmPRA shows a similar tendency. For the three filtering strategies the precision for the larger PRA is always better or equal than the precision of the smaller PRA.

3.5 Related Work

Since the task of using PRA to align ontologies was introduced as task #4 in the Anatomy track in OAEI 2008, eight ontology alignment systems have participated (as listed in Table 3.8), including SAMBO and SAMBOdtf (for which we used fPRA and dtfPRA respectively). Our work in this chapter, published as [14] before OAEI 2009, is the first experimental study on the different uses of PRAs to align ontologies.

	2008	2009	2010
SAMBO	✓	-	-
SAMBOdtf	✓	-	-
RiMOM	✓	-	-
ASMOV	✓	✓	✓
Anchor-Flood	-	✓	-
TaxoMap	-	✓	-
AgreementMaker	-	✓	✓
CODI	-	-	✓

Table 3.8: Overview of participants of the OAEI Anatomy task #4 from 2008 to 2010.

As a participant in 2008, RiMOM [85] is an ontology alignment system utilizing several alignment strategies based on specific ontology information. The selection and combination of these strategies are determined automatically based on the estimation of three ontology feature factors. In its paper for OAEI [86], the use of the PRA is not explained. However, based on the description in [85], it is likely that the mappings in the PRA are used at the similarity propagation phase to calculate structural similarities.

ASMOV [87] uses a set of matching strategies considering lexical description, external structure, internal structure and individuals in the ontologies. Based on the calculated similarities, the alignment result is extracted and refined iteratively through a semantic verification process, where mappings causing semantic inferences unsupported by the ontologies are removed from the result. Having the PRA as an optional input, ASMOV adopts a similar strategy as fPRA, by making the mappings in the PRA supersede others during the alignment extraction phase.

With the focus on reducing the computational time, Anchor-Flood [88] starts off with anchors, which are pairs of similar concepts already aligned. By collecting the neighbors of these anchors in the concept taxonomy gradually, the algorithm produces a segmented alignment between the corresponding neighbor concepts across ontologies. Though the use of the PRA is not explained in [89], it is likely that the mappings in the PRA are used as anchors according to the description in [88].

TaxoMap [90] is an alignment tool performing oriented alignment, which takes into account labels and sub-class descriptions in ontologies. It employs the PBM [91] algorithm, allowing the use of some predefined equivalence mappings to partition the ontologies into pairs of mappable blocks (or modules). For the task #4 in Anatomy track, the mappings in the PRA are used as predefined mappings to partition the ontologies [92], sharing a similar intuition with mgPRA.

AgreementMaker [93] is an iterative ontology alignment system capable of executing, evaluating, comparing and combining multiple matching strategies. It uses the PRA in two different ways, partitioning the ontologies into mappable parts and filtering mappings which are considered incorrect regarding some structural constraints [94].

CODI [95] is a probabilistic-logical alignment system, based on the Markov logic. It transforms the alignment problem to a maximum-a-posteriori optimization problem, where the mappings in the PRA could be directly added and used for retaining the good correspondences and excluding contradicting ones.

System in 2008	Δ-Precision	Δ-Recall	Δ-F-Measure
SAMBO	+0.024 0.636→0.660	−0.002 0.626→0.624	+0.011 0.631→0.642
SAMBOdtf	+0.040 0.563→0.603	+0.008 0.622→0.630	+0.025 0.591→0.616
ASMOV	+0.063 0.339→0.402	−0.004 0.258→0.254	+0.019 0.293→0.312
RiMOM	+0.012 0.700→0.712	+0.000 0.370→0.370	+0.003 0.484→0.487

Table 3.9: Results in 2008 [80], including changes in precision, recall and F-measure based on comparing $M_1 \backslash R_p$ resp. $M_4 \backslash R_p$ with the unknown part of the reference alignment $R \backslash R_p$. M_1 and M_4 are respectively the system alignment result without and with using the PRA, while R_p and R respectively represent the PRA and RA.

System in 2009	Δ-Precision	Δ-Recall	Δ-F-Measure
SAMBOdtf_{2008}	+0.020 0.837→0.856	+0.003 0.867→0.870	+0.011 0.852→0.863
ASMOV	+0.034 0.759→0.792	−0.018 0.808→0.790	+0.009 0.782→0.791
Anchor-Flood	+0.005 0.838→0.843	+0.003 0.825→0.827	+0.004 0.831→0.835
TaxoMap	+0.019 0.878→0.897	−0.026 0.732→0.706	−0.008 0.798→0.790
AgreementMaker	+0.128 0.870→0.998	−0.181 0.831→0.650	−0.063 0.850→0.787

Table 3.10: Results in 2009 [12], including changes in precision, recall and F-measure based on comparing $M_1 \cup R_p$ resp. $M_4 \cup R_p$ against reference alignment R. Note that the evaluation criteria are different from the previous year, while the meanings of M_1, M_4, R_p and R are the same as before.

Tables 3.9, 3.10 and 3.11 show the official results of participating systems on task #4 in the Anatomy track in OAEI 2008, 2009 and 2010 respectively.[7] As explained in the captions of the tables, there were some differences in the evaluation criteria or the data across years. We can notice that almost all participating systems profit from the use of the PRA by an increased precision with respect to the reference alignment. However, only SAMBOdtf, Anchor-Flood and CODI have an increased recall.

Even though there is no other work specialized on the use of PRAs for the ontology alignment, we found that some existing ontology alignment systems allow for the use of PRAs. This means that these algorithms use a set of mappings as input, although they may not always have been validated. Some systems make use of

[7] Representing the SAMBOdtf result in 2008, SAMBOdtf_{2008} is added in the results of 2009 and 2010 by the organizers of OAEI Anatomy track, as a reference of the best results measured in the last years.

System in 2010	Δ-Precision	Δ-Recall	Δ-F-Measure
AgreementMaker	$+0.025$ 0.904→0.929	-0.025 0.876→0.851	-0.002 0.890→0.888
ASMOV	$+0.029$ 0.808→0.837	-0.016 0.824→0.808	$+0.006$ 0.816→0.822
CODI	-0.002 0.970→0.968	$+0.030$ 0.716→0.746	$+0.019$ 0.824→0.843
SAMBOdtf_{2008}	$+0.021$ 0.837→0.856	$+0.003$ 0.867→0.870	$+0.011$ 0.852→0.863

Table 3.11: Results in 2010 [96], including changes in precision, recall and F-measure based on comparing $M_1 \cup R_p$ resp. $M_4 \cup R_p$ against reference alignment R. The meanings of M_1, M_4, R_p and R are the same as before. However, this time, both the ontology files and the RA are slightly modified so as to get rid of incoherence encountered in the last years.

anchors to partition large-scale ontologies into similar blocks before aligning (e.g. Falcon-AO [97]), or propagating similarities along the structure of ontologies (e.g. Anchor-PROMPT [98] and Similarity-Flooding [99]), where anchors are usually a set of mappings generated by the basic linguistic matchers which could be imprecise. Some systems also generate and take advantage of anchors, but after the user validation, to initialize similarities prior to the structural matching (e.g. Cupid [100]), or refine the alignment results during an iterative alignment process (e.g. S-Match [101] and SAMBO [13]). For these systems, the mappings in the PRAs could be added to or used as the anchors to achieve better results. For algorithms which recommend ontology alignment strategies, PRAs could also be used to setup the training data for learning the weights of a matching scheme [102] or evaluating different alignment strategies [103]. However, as admitted by the organizers of OAEI Anatomy track, depending on the characteristic of the ontologies and the PRA, sometimes it is a challenging task to use the PRA in an appropriate way [12].

3.6 Summary

In this chapter we have investigated whether and how a PRA can be used in ontology alignment by experimenting with using a PRA in the different components of ontology alignment systems. The use of PRA in preprocessing and filtering reduces the number of suggestions and in most cases leads to an improvement in precision. In some cases also the recall improved. Filter with PRA should always be used. For approaches using structural information the quality of the structure in the underlying ontologies has a large impact. The matcher using linguistic patterns in the PRA mappings can be used for finding new suggestions. The differences between the results for the algorithms that use a PRA and the base systems are relatively small. However, considering the nature of the test cases and the fact that SAMBO and SAMBOdtf perform already well on their own, even small improvements are valuable. Also, for the large test case, due to the choice of the PRA all newly found mappings are non-trivial.

Chapter 4

Debugging Missing is-a Structure within Networked Ontologies

In this chapter, we address the problem of missing structure, in particular missing is-a relations, in networked ontologies, where the missing is-a relations are considered as a kind of modeling defects in ontologies. Assuming that ontologies are networked by PRAs (i.e. all the mappings between the networked ontologies are correct) and all the existing is-a relations in the ontologies are correct, we propose an ontology debugging approach to detect and repair the missing is-a relations.

4.1 Introduction

As a basic language primitive in ontologies, the structural relations (is-a hierarchy) have a significant meaning for the modeling of a domain. By organizing the loose terms in the ontology into a taxonomy, the is-a structure not only explains the relationships between these terms, but also makes the meaning of terms more explicit within the context. Therefore, many ontologies are built up as taxonomies, and the structural information is widely used in semantically-enabled applications, such as ontology-based search and annotation. The is-a structure is also an important information in ontology engineering research. For instance, most current ontology alignment systems use structure-based strategies to find mappings between the terms in different ontologies (e.g. overview in [10]) and the modeling defects in the structure of the ontologies have an important influence on the quality of the ontology alignment results [12].

However, as the ontologies grow in size, it is difficult to ensure the correctness and completeness of the structure of the ontologies. Some structural relations may be missing or some existing or derivable relations may be unintended. Detecting and resolving these defects requires, in contrast to semantic defects, the use of domain knowledge. One interesting kind of domain knowledge are the other ontologies and information about connections between these ontologies. For instance, in the case of the Anatomy track in the 2008 and 2009 Ontology Alignment Evaluation Initiative (OAEI) two ontologies, Adult Mouse Anatomy Dictionary (MA[1], 2744 concepts) and the NCI Thesaurus anatomy (NCI-A[2], 3304 concepts), and a PRA containing 988 mappings are given. Using one ontology and the mappings as domain knowledge for the other ontology (and vice versa), it was shown in [14] that at least 121 is-a relations in MA and 83 in NCI-A are missing and should be repaired. This is not an uncommon case. It is well-known that people that are not expert in knowledge representation often misuse and confuse equivalence, is-a and part-of (e.g. [11]), which leads to problems in the structure of the ontologies.

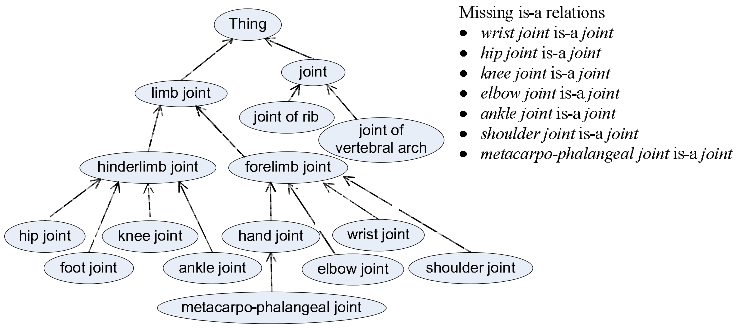

Figure 4.1: A part of MA regarding the concept *joint*.

Once the missing is-a relations are found, the structure of the ontology can be repaired by adding the missing is-a relations themselves, but this is not always the most interesting solution for a domain expert. For instance, Figure 4.1 shows a part of MA regarding the concept *joint* (is-a relations shown with arrows). Using NCI-A and the PRA as domain knowledge, 7 missing is-a relations are found. The ontology could be repaired by adding these missing is-a relations themselves. However, for the missing is-a relation "*wrist joint* is-a *joint*", knowing that there is an is-a relation between *wrist joint* and *limb joint*, a domain expert will most likely prefer to add the is-a relation "*limb joint* is-a *joint*" instead. This is correct from a modeling perspective as well as more informative and would lead to the fact that the missing is-a relation between *wrist joint* and *joint* can be derived. In this particular case, using "*limb joint* is-a *joint*" would actually also lead to the repairing of the other 6 missing is-a relations, as well as others that were not found before (e.g. "*hand joint* is-a *joint*"). In general, such a decision should be made by domain experts.

In this chapter, we deal with detecting and repairing the missing is-a structure in ontologies in the context of domain knowledge represented by the ontology net-

[1] http://www.informatics.jax.org/searches/AMA_form.shtml
[2] http://www.cancer.gov/cancerinfo/terminologyresources/

work. Assuming that the existing is-a relations in the ontologies are correct, as well as the mappings in the PRAs, we use them as domain knowledge to detect the missing is-a relations in these ontologies. Then, we develop algorithms to generate and recommend possible ways of repairing, which are relevant for domain experts, as well as algorithms to rank missing is-a relations and execute the repairing. Based on these, we develop the system RepOSE (*Rep*air of *O*ntological *S*tructure *E*nvironment), which allows a domain expert to debug the missing is-a structure of ontologies in a semi-automatic way.

The remainder of this chapter is organized as follows. In section 4.2 we present the theory for our debugging approach. The overview of the whole process is given in section 4.3. Section 4.4 introduces the algorithms for the detection process, while section 4.5 explains the algorithms for the repairing process which involves generating repairing actions, ranking missing is-a relations, recommending and executing repairing actions. Our system RepOSE and its use are described in section 4.6. Further, we discuss an experiment in section 4.7. Related work is presented in section 4.8 and the chapter is summarized in section 4.9.

4.2 Theory

Our approach for debugging missing is-a relations in an ontology network contains two parts, i.e. detecting and repairing. The former deals with the identification of the missing is-a relations in the networked ontologies, while the latter deals with repairing the structure of the ontologies.

4.2.1 Preliminaries

The setting that we study is the case where the ontologies are defined using named concepts and subsumption axioms. Most ontologies contain this case and many of the most well-known and used ontologies, e.g. in the life sciences, are covered by this setting.

Definition 4.2.1. *An ontology* O *is represented by a tuple* (C,I) *with* C *its set of named concepts and* $\mathrm{I} \subseteq \mathrm{C} \times \mathrm{C}$ *a representation of its is-a structure.*

A PRA between two ontologies contains a set of correct mappings between the concepts of different ontologies.

Definition 4.2.2. *Let* O_i *and* O_j *be two ontologies, a* ***Partial Reference Alignment (PRA)*** *between* O_i *and* O_j *is a set of correct mappings such that each mapping is a 3-tuple* (c_i, c_j, r)*, where* c_i *is a concept in* O_i*,* c_j *is a concept in* O_j*, and r specifies the semantic relation between* c_i *and* c_j*.*

We assume that concepts can participate in multiple mappings. Further, for a PRA between O_i and O_j, there is a corresponding PRA between O_j and O_i, such that there is a mapping (c_i, c_j, r) in the former iff there is a corresponding mapping (c_j, c_i, r^{-1}) in the latter, where r^{-1} denotes the inverse relation of r. For the semantic relation r, we currently consider *equivalent* ($\equiv$), *subsumed-by* ($\rightarrow$) and *subsumes* ($\leftarrow$), whose inverse relations are *equivalent*, *subsumes* and *subsumed-by* respectively. In this way, a PRA could be represented by a set of is-a relations (or subsumption axioms).

Definition 4.2.3. *The **representation for a PRA** between* O_i *and* O_j*, denoted by* P_{ij}*, is a set of pairs representing is-a relations, such that for each mapping in the PRA:* $(c_i, c_j, \rightarrow)$ *is represented by* (c_i, c_j) *in* P_{ij}*;* $(c_i, c_j, \leftarrow)$ *is represented by* (c_j, c_i)*; and* $(c_i, c_j, \equiv)$ *is represented by both* (c_i, c_j) *and* (c_j, c_i)*.*

An ontology network is defined as follows.

Definition 4.2.4. *An **ontology network*** N *is a tuple* $(\mathbb{O}, \mathbb{P})$ *with* $\mathbb{O} = \{\mathrm{O}_k\}_{k=1}^n$ *the set of the ontologies in the network (called the networked ontologies) and* $\mathbb{P} = \{\mathrm{P}_{ij}\}_{i,j=1;i<j}^n$ *the set of representations for the PRAs between these ontologies.*

The domain knowledge of an ontology network is represented by the *induced ontology* defined below.

Definition 4.2.5. *Given an ontology network* $\mathrm{N} = (\mathbb{O}, \mathbb{P})$ *where* $\mathbb{O} = \{\mathrm{O}_k\}_{k=1}^n$ *and* $\mathbb{P} = \{\mathrm{P}_{ij}\}_{i,j=1;i<j}^n$ *and for every k,* $\mathrm{O}_k = (\mathrm{C}_k, \mathrm{I}_k)$*, the **induced ontology*** O_N *for* N *is an ontology* $\mathrm{O}_N = (\mathrm{C}_N, \mathrm{I}_N)$ *such that* $\mathrm{C}_N = \cup_{k=1}^n \mathrm{C}_k$ *and* $\mathrm{I}_N = \cup_{k=1}^n \mathrm{I}_k \cup \cup_{i,j=1;i<j}^n \mathrm{P}_{ij}$*.*

This definition states that, the induced ontology for an ontology network is an ontology, whose concepts are the concepts of the ontologies in the network, and its is-a relations are the is-a relations of the ontologies together with the is-a relations representing the mappings of the PRAs in the network.

4.2.2 Theory for detecting

Given an ontology network, assuming that all existing is-a relations in the ontologies are correct, we use the domain knowledge of the ontology network to detect the missing is-a relations in these networked ontologies. For each ontology in the network, the set of missing is-a relations derivable from the ontology network consists of is-a relations between two concepts of the ontology, which can be inferred using logical derivation from the induced ontology of the network, but not from the networked ontology alone.

Definition 4.2.6. *Given an ontology network* $\mathrm{N} = (\mathbb{O}, \mathbb{P})$ *and an ontology* $\mathrm{O}_k = (\mathrm{C}_k, \mathrm{I}_k) \in \mathbb{O}$*, the set of **missing is-a relations for the networked ontology*** O_k ***derivable from the ontology network*** N*, denoted by* M_k*, is the set of is-a relations such that* $\mathrm{M}_k = \{(a,b) \in \mathrm{C}_k \times \mathrm{C}_k \mid \mathrm{O}_N \models a \rightarrow b \wedge \mathrm{O}_k \not\models a \rightarrow b\}$*.*

Definition 4.2.7. *Given an ontology network* $\mathrm{N} = (\mathbb{O}, \mathbb{P})$ *where* $\mathbb{O} = \{\mathrm{O}_k\}_{k=1}^n$*, the set of **missing is-a relations for the networked ontologies*** $\mathbb{O}$ ***derivable from the ontology network*** N*, denoted by* M_N*, is the set of is-a relations such that* $\mathrm{M}_N = \cup_{k=1}^n \mathrm{M}_k$*.*

As an example, consider the ontology network in Figure 4.2. It contains two ontologies with their is-a hierarchies (marked by the solid arrows), which are related via 3 mappings with equivalence relations (marked by the dashed lines). According to the definition above, there are two missing is-a relations derivable from the network, $(ankle\_joint_1, joint_1)$ in ontology 1 and $(ankle\_joint_2, limb\_joint_2)$ in ontology 2 (marked by the dashed arrows). Domain experts may argue that some is-a relations, such as $(knee\_joint_1, joint_1)$ and $(hip\_joint_2, limb\_joint_2)$, are also missing is-a relations. However, these cannot be found using logical derivation within the network and are thus not missing is-a relations *derivable* from the network as defined in Definition 4.2.6 and 4.2.7.

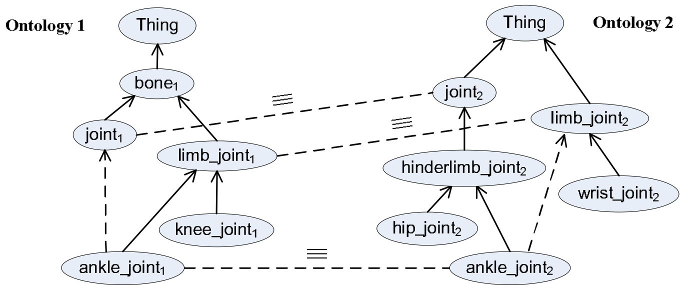

Figure 4.2: An example ontology network.

From now on, in this chapter, whenever missing is-a relations are mentioned, we mean the missing is-a relations derivable from the network.

4.2.3 Theory for repairing

Our goal for repairing is to repair the original ontologies by adding a set of is-a relations, called a *structural repair*, to each ontology such that the missing is-a relations can be derived from the ontology extended with the newly-added is-a relations. Therefore, the structural repair only contains is-a relations between concepts within the same ontology.

Definition 4.2.8. *Let* M_k *be the set of missing is-a relations for an ontology* $\mathrm{O}_k = (\mathrm{C}_k, \mathrm{I}_k)$ *in an ontology network* N*. A* ***structural repair for the networked ontology*** O_k *with respect to* M_k*, denoted by* R_k*, is a set of is-a relations such that* $\mathrm{R}_k \subseteq \mathrm{C}_k \times \mathrm{C}_k$ *and for each missing is-a relation* $(a,b) \in \mathrm{M}_k$*,* $(\mathrm{C}_k, \mathrm{I}_k \cup \mathrm{R}_k) \models a \to b$*.*

The elements in a structural repair are called *repairing actions*. Accordingly, we have the definition of a structural repair for the networked ontologies.

Definition 4.2.9. *Let* $\mathrm{M}_N = \cup_{k=1}^{n} \mathrm{M}_k$ *be the set of missing is-a relations for an ontology network* $\mathrm{N} = (\mathbb{O}, \mathbb{P})$ *where* $\mathbb{O} = \{\mathrm{O}_k\}_{k=1}^{n}$ *and* M_k *the set of missing is-a relations for* O_k *derivable from* N*. A* ***structural repair for the networked ontologies*** $\mathbb{O}$ *with respect to* M_N*, denoted by* R_N*, is a set of is-a relations such that* $\mathrm{R}_N = \cup_{k=1}^{n} \mathrm{R}_k$*, where* R_k *is a structural repair for* O_k *with respect to* M_k*.*

An immediate consequence of the definition is that, for the networked ontologies, the set of missing is-a relations is in itself a structural repair. Another consequence is that adding is-a relations between concepts of any single ontology in the network to a structural repair for the networked ontologies also constitutes a structural repair.

As mentioned in Section 4.1, not all structural repairs are equally useful or interesting for a domain expert. Therefore, we define three preference relations between structural repairs for the networked ontologies.

The first preference relation prefers structural repairs which do not contain non-contributing is-a relations for repairing. For example, in the case of Figure 4.2, the structural repair $\{(ankle\_joint_1, joint_1), (ankle\_joint_2, limb\_joint_2)\}$ is preferred to $\{(ankle\_joint_1, joint_1), (knee\_joint_1, joint_1), (ankle\_joint_2, limb\_joint_2)\}$, since the repairing action $(knee\_joint_1, joint_1)$ is non-contributing for repairing the missing is-a relations.

Definition 4.2.10. *Given an ontology network* $\mathrm{N} = (\mathbb{O}, \mathbb{P})$*, let* R_N *and* R'_N *be structural repairs for the networked ontologies* $\mathbb{O}$ *with respect to* M_N*, then* R_N *is* ***axiom-preferred*** *to* R'_N *(notation* $\mathrm{R}_N \ll_A \mathrm{R}'_N$*) iff* $\mathrm{R}_N \subseteq \mathrm{R}'_N$.

The set of missing is-a relations is not always the most interesting structural repair for the domain expert. For instance, in the case of Figure 4.2, the structural repair $\{(limb\_joint_1, joint_1), (ankle\_joint_2, limb\_joint_2)\}$ is, for a domain expert, a more preferred way to repair the ontologies than the structural repair $\{(ankle\_joint_1, joint_1), (ankle\_joint_2, limb\_joint_2)\}$ which only contains the missing is-a relations. The former also repairs the ontologies, is correct according to the domain and is more informative. By using more informative repairing actions, we are able to add more (and sometimes previously unknown) knowledge to our ontology. For instance, by introducing $limb\_joint_1 \rightarrow joint_1$, we have also introduced the correct facts $ankle\_joint_1 \rightarrow joint_1$ (the missing is-a relation) as well as $knee\_joint_1 \rightarrow joint_1$ (which was also missing, but could not be derived from the network). Our second preference relation prefers to use as informative is-a relations as possible for repairing.

Definition 4.2.11. *Let* (x_1, y_1) *and* (x_2, y_2) *be two different repairing actions for the same ontology* O *(i.e.* $x_1 \not\equiv x_2$ *or* $y_1 \not\equiv y_2$*), then we say that* (x_1, y_1) *is more* ***informative*** *than* (x_2, y_2) *iff* $\mathrm{O} \models x_2 \rightarrow x_1 \wedge y_1 \rightarrow y_2$.

Definition 4.2.12. *Given an ontology network* $\mathrm{N} = (\mathbb{O}, \mathbb{P})$*, let* R_N *and* R'_N *be structural repairs for the networked ontologies* $\mathbb{O}$ *with respect to* M_N*. Then* R_N *is* ***information-preferred*** *to* R'_N *(notation* $\mathrm{R}_N \ll_I \mathrm{R}'_N$*) iff* $\exists\ (x_1, y_1) \in \mathrm{R}_N$*,* $(x_2, y_2) \in \mathrm{R}'_N$*:* (x_1, y_1) *is more informative than* (x_2, y_2)*.*

Further, some structural repairs may introduce equivalence relations between concepts in some ontology which are only connected by an is-a relation in the original ontology. For example, in the case of Figure 4.2, the structural repair $\{(bone_1, joint_1), (ankle\_joint_2, limb\_joint_2)\}$ will change the original is-a relation $(joint_1, bone_1)$ in ontology 1 into an equivalence relation. Although such a structural repair may result in a consistent ontology, this is usually not desired from a modeling perspective. The third preference relation prefers not to change is-a relations in an original ontology into equivalence relations.

Definition 4.2.13. *Given an ontology network* $N = (\mathbb{O}, \mathbb{P})$*, let* $R_N = \cup_{k=1}^{n} R_k$ *and* $R'_N = \cup_{k=1}^{n} R'_k$ *be structural repairs for the networked ontologies* $\mathbb{O} = \{O_k\}_{k=1}^{n}$ *with respect to* M_N*, where for every k,* $O_k = (C_k, I_k)$*. Then* R_N *is* ***strict-hierarchy-preferred*** *to* R'_N *(notation* $R_N \ll_{SH} R'_N$*) iff* $\exists O_i \in \mathbb{O}$ *and* $(a,b) \in I_i : O_i \not\models a \equiv b$ *and* $(C_i, I_i \cup R_i) \not\models a \equiv b$ *and* $(C_i, I_i \cup R'_i) \models a \equiv b$*.*

We note that, according to our definitions, it is possible that one structural repair is preferred to a second structural repair, while at the same time the second structural repair is preferred to the first one. For example, in the case of Figure 4.2, let R_1 be the structural repair $\{(limb\_joint_1, joint_1), (ankle\_joint_2, limb\_joint_2)\}$ and R_2 be the structural repair $\{(ankle\_joint_1, joint_1), (hinderlimb\_joint_2, limb\_joint_2)\}$. Then $R_1 \ll_I R_2$ and $R_2 \ll_I R_1$. The first preference is based on the fact that $(limb\_joint_1, joint_1)$ is more informative than $(ankle\_joint_1, joint_1)$, while the second preference is based on the fact that $(hinderlimb\_joint_2, limb\_joint_2)$ is more informative than $(ankle\_joint_2, limb\_joint_2)$. In this case it is, however, possible to find a third structural repair, e.g. $\{(limb\_joint_1, joint_1), (hinderlimb\_joint_2, limb\_joint_2)\}$, that is strictly more information-preferred than both.

In general, we would want structural repairs that are maximally preferred.

Definition 4.2.14. *Given an ontology network* $N = (\mathbb{O}, \mathbb{P})$*, a structural repair* R_N *for the networked ontologies* $\mathbb{O}$ *with respect to* M_N*, is maximally preferred with respect to the preference relation* $\ll$ *iff for all structural repairs* R'_N *for* $\mathbb{O}$ *with respect to* M_N *it holds that if* $R'_N \ll R_N$ *then* $R_N \ll R'_N$*.*

4.3 Overview of the approach

In this section, we give an overview of our debugging approach. As illustrated in Figure 4.3, the process consists of 5 phases and is driven by a domain expert. The input is a set of ontologies networked by a set of PRAs.

The user starts with detecting missing is-a relations for all the networked ontologies **(Phase 1)**.The algorithm for detecting missing is-a relations is described in Section 4.4.

A naive way of repairing would be to compute all possible structural repairs for the networked ontologies. This is in practice infeasible as it involves all the ontologies and all the missing is-a relations in the network. It is also hard for domain experts to choose between structural repairs containing large sets of repairing actions for all the ontologies. Therefore, in our approach, we repair ontologies one at a time. After one ontology is chosen for repairing, we generate a set of possible repairing actions for each missing is-a relation in the chosen ontology **(Phase 2)** so that the user can repair the missing is-a relations one by one. The algorithm for generating possible repairing actions takes into account the preferences defined in section 4.2.3. In general, there will be many missing is-a relations that need to be repaired and some of them may be easier to start with such as the ones with fewer

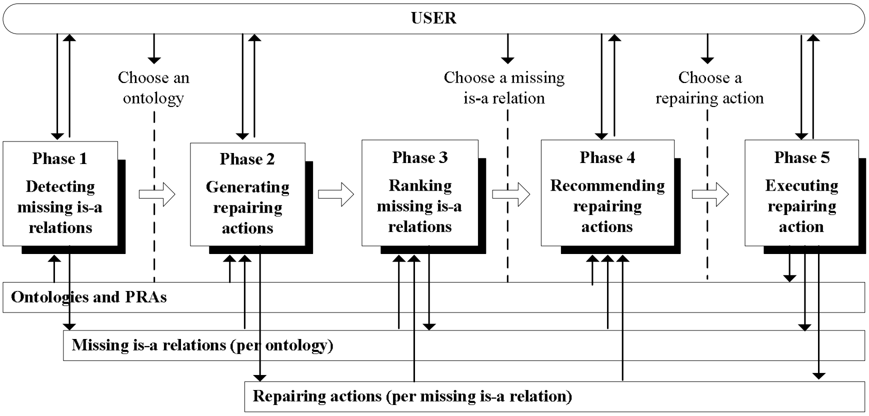

Figure 4.3: Approach for debugging missing is-a structure in networked ontologies.

repairing actions. We therefore rank them with respect to the number of possible repairing actions **(Phase 3)**.

After this, the user can select a missing is-a relation to repair and choose between possible repairing actions. To facilitate this process, we developed a method to recommend the most informative repairing actions based on domain knowledge **(Phase 4)**. Once the user chooses a repairing action to execute, the chosen repairing action is then added to the ontology and the consequences are computed **(Phase 5)**. Some missing is-a relations may be repaired by the executed repairing action. Some missing is-a relations may have their repairing actions changed. Further, some new missing is-a relations may be found.

At any time during the process, the user can switch the ontology to repair or start earlier phases.

4.4 Detecting the missing is-a relations in networked ontologies

The missing is-a relations derivable from the network could be found by checking the is-a relations between all concepts in every single ontology. If an is-a relation is not derivable from the ontology but derivable from the network, it is a missing is-a relation. However, some of these missing is-a relations are redundant in the sense that they can be repaired by the repairing of other missing is-a relations. It can be shown that only the missing is-a relations whose concepts appear in the mappings of the PRAs are necessary for repairing. (As a shorthand, we call the concepts appearing in the mappings of the PRAs *PRA concepts*.)

Proposition 4.4.1. *For each missing is-a relation in the network, there must exist a missing is-a relation whose concepts are PRA concepts, such that the repairing* *of the latter also repairs the former.*

Proof. Suppose in an ontology network N as defined in definition 4.2.4, there is a missing is-a relation (a,b) in an ontology O. According to the definition 4.2.6, the relation $a \rightarrow b$ is not derivable from O but derivable from the ontology network. So, there must exist at least one concept from another ontology in the network, for instance z, such that $O_N \models a \rightarrow z \rightarrow b$. Because concepts a and z reside in different ontologies, the relation $a \rightarrow z$ must be supported by a mapping between a concept in O and a concept in another ontology in the network, for instance $x \rightarrow x'$ (or $x \equiv x'$), satisfying $O_N \models a \rightarrow x \rightarrow x' \rightarrow z$, where x is a PRA concept in ontology O. Likewise, for concepts z and b, the relation $z \rightarrow b$ must also be supported by a mapping between a concept in O and a concept in another ontology in the network, for instance $y' \rightarrow y$ (or $y' \equiv y$), satisfying $O_N \models z \rightarrow y' \rightarrow y \rightarrow b$, where y is a PRA concept in ontology O. We can then deduce that $x \rightarrow y$ is derivable from the ontology network because $O_N \models a \rightarrow x \rightarrow x' \rightarrow z \rightarrow y' \rightarrow y \rightarrow b$. Since $a \rightarrow b$ is not inferrable from O, the relation $x \rightarrow y$ can not be inferred from O either. This means that (x,y) is also a missing is-a relation in the network, and the repairing of missing is-a relation (x,y) also repairs (a,b).

□

Based on proposition 4.4.1, repairing the missing is-a relations between PRA concepts also repairs all other missing is-a relations derivable from the ontology network. Therefore, our algorithm in Figure 4.4 considers only missing is-a relations between PRA concepts.

Input
The ontology network $N = (\mathbb{O}, \mathbb{P})$, the induced ontology O_N,
the set of is-a relations to check $M_N^\star$.
Output
Missing is-a relations M_N, the updated set of is-a relations to check $M_N^\star$.
Algorithm
For each ontology $O \in \mathbb{O}$ where $O = (C, I)$:
  For every $(a,b) \in M_N^\star$ and $a, b \in C$:
    If $O \models a \rightarrow b$ then remove (a,b) from $M_N^\star$;
    If $O \not\models a \rightarrow b$ and $O_N \models a \rightarrow b$ then:
      Add (a,b) as a missing is-a relation to M_N;
      Remove (a,b) from $M_N^\star$.

Figure 4.4: Algorithm for detecting missing is-a relations.

In the algorithm in Figure 4.4, the global variable $M_N^\star$ represents the set of is-relations which we need to check to find missing is-a relations in the network. Before the algorithm is run for the first time, $M_N^\star$ is initialized to be the set of pairs (a,b) where a and b are PRA concepts in the same ontology. For each element (a,b) in $M_N^\star$ we then check whether $a \rightarrow b$ can be derived in the ontology to which

a and b belong. If so, then this is not a missing is-a relation and (a,b) is removed from $M_N^\star$. Otherwise, we check whether $a \to b$ can be derived from the network. If so, then it is a missing is-a relation and we add (a,b) to M_N, representing the set of missing is-a relations, and remove it from $M_N^\star$. Otherwise, $a \to b$ can neither be derived from the ontology nor from the network. It then remains in $M_N^\star$ as it may become derivable later when we have repaired part of the network. As all pairs of PRA concepts are checked, our algorithm ensures that all missing is-a relations between PRA concepts that can be derived from the current network will be found.

After the missing is-a relations are found, they will be repaired in later phases of the debugging process and this will bring changes to the is-a structures of the repaired ontologies and the induced ontology. Therefore, it is possible that some new is-a relations become derivable from the network and thus generate new missing is-a relations. For example, in the case of Figure 4.2, suppose we repair $(ankle\_joint_1, joint_1)$ by adding the is-a relation $(limb\_joint_1, joint_1)$ in ontology 1. Then, when re-running the detection algorithm, we find a new missing is-a relation $(limb\_joint_2, joint_2)$, since $(limb\_joint_2, joint_2)$ has now become inferrable from the induced ontology and it is still not inferrable from ontology 2. Therefore, after executing repairing actions, we need to re-run the detection algorithm to find new missing is-a relations. As we do not change the mappings between the ontologies when repairing, the set of PRA concepts does not change. Therefore, there are no additions to $M_N^\star$ and the initial value of $M_N^\star$ when re-running the detection algorithm is the same as the final value of $M_N^\star$ in the previous run.

4.5 Repairing the missing is-a relations in networked ontologies

As explained in Section 4.3, our approach deals with the networked ontologies one at a time. For the ontology under repair, possible repairing actions are generated for all its missing is-a relations. After ranking these missing is-a relations, the user can select a missing is-a relation to repair, and we provide an algorithm that recommends repairing actions. Further, we developed an algorithm that, upon the repairing of a missing is-a relation, detects for which missing is-a relations the set of repairing actions needs to be updated, and updates these.

4.5.1 Generating repairing actions

Basic algorithm

In our basic algorithm (see Figure 4.5), when generating repairing actions for a missing is-a relation, we take into consideration that all missing is-a relations will be repaired (least informative repairing action), but we do not take into account the consequences of the actual (possibly more informative) repairing actions that will be performed for other missing is-a relations.

In this algorithm, we store the ontology in a knowledge base and add the missing is-a relations. As we know that these missing is-a relations are derivable from

Input
The ontology under repair O, its set of missing is-a relations M.
Output
Repairing actions.
Algorithm
1. Initialize KB with ontology;
2. For every missing is-a relation $(a,b) \in$ M: add the axiom $a \rightarrow b$ to the KB;
3. For each $(a,b) \in$ M:
   $Source(a,b)$:= super-concepts(a) $-$ super-concepts(b);
   $Target(a,b)$:= sub-concepts(b) $-$ sub-concepts(a);
4. Missing is-a relation (a,b) can be repaired by choosing an element from $Source(a,b) \times Target(a,b)$.

Figure 4.5: The basic algorithm for generating repairing actions.

the network, adding them will introduce the desired connections. It guarantees that, for the ontology under repair, all inferrable is-a relations between its concepts in the network will also become inferrable from the ontology. Essentially, this conforms to a structural repair containing the least informative repairing actions for each of the missing is-a relations in the ontology. Then, for each missing is-a relation, we generate its possible repairing actions by computing two sets of concepts, called Source and Target sets. A possible repairing action regarding missing is-a relation (a,b) is an is-a relation (s,t) where s is an element from its Source set and t is an element from its Target set. The computation of Source and Target sets ensures that we only compute repairing actions that contribute to repairing the missing is-a relation (a,b) (preference $\ll_A$ in definition 4.2.10), and every possible repairing action (s,t) satisfies $a \rightarrow s$ and $t \rightarrow b$ (preference $\ll_I$ in definition 4.2.12). At the same time, it is guaranteed that repairing action (a,t) or (s,b) will not introduce new equivalence relations, where in the source ontology we have only is-a relations (preference $\ll_{SH}$ in definition 4.2.13).

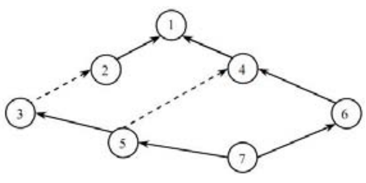

Figure 4.6: Example 1.

As an example, consider the case presented in Figure 4.6, where O $= (\text{C},\text{I})$ is an ontology with concepts C $= \{1,2,3,4,5,6,7\}$ and is-a relations (shown in full lines in Figure 4.6) I $= \{(7,5),(7,6),(5,3),(2,1),(6,4),(4,1)\}$. (I represents the is-a hierarchy and thus also all is-a relations derived from the elements in I.) The set of missing is-a relations (shown in dashed lines in Figure 4.6) is M $= \{(5,4),(3,2)\}$.

The algorithm then generates the following Source and Target sets.

$Source(5,4) = \{5,3,2,1,4\} - \{4,1\} = \{5,3,2\}$
$Target(5,4) = \{4,6,7,5\} - \{5,7\} = \{4,6\}$
$Source(3,2) = \{3,2,1\} - \{2,1\} = \{3\}$
$Target(3,2) = \{2,3,5,7\} - \{3,5,7\} = \{2\}$

For missing is-a relation $(3,2)$ the only generated repairing action is $(3,2)$. For missing is-a relation $(5,4)$ any of the repairing actions $(5,4)$, $(5,6)$, $(3,4)$, $(3,6)$, $(2,4)$, $(2,6)$ together with (any of) the generated repairing action(s) for $(3,2)$ leads to the derivation of the missing is-a relation $(5,4)$ in the extended ontology. The example also shows the importance of initially adding the missing is-a relations to the knowledge base. The possible repairing action $(2,4)$ for missing is-a relation $(5,4)$ would not be generated when we do not take into account that missing is-a relation $(3,2)$ will be repaired.[3] Further, the example also shows that we do not introduce repairing actions that would turn is-a relations in the original ontology into equivalence relations. For instance, adding $(1,4)$ would lead to the fact that missing is-a relation $(5,4)$ would be derivable in the extended ontology, but also leads to making 1 and 4 equivalent.

Input
The ontology under repair O, its set of missing is-a relations M.
Output
Repairing actions.
Algorithm
1. Initialize KB with ontology ;
2. For every missing is-a relation $(a,b) \in$ M:
    Create two new concepts x and y in the KB;
    Add the axioms $a \rightarrow x$, $x \rightarrow y$, $y \rightarrow b$ to the KB;
3. For each $(a,b) \in$ M:
    $Source\text{-}ext(a,b)$:= super-concepts(a) $-$ super-concepts(x);
    $Target\text{-}ext(a,b)$:= sub-concepts(b) $-$ sub-concepts(y);
4. Missing is-a relation (a,b) can be repaired by choosing an original ontology element from $Source\text{-}ext(a,b)$ and an original ontology element from $Target\text{-}ext(a,b)$.

Figure 4.7: The extended algorithm for generating repairing actions.

Extended algorithm

Our extended algorithm (see Figure 4.7) for finding repairing actions for a particular missing is-a relation takes into account influences of other missing is-a relations that are valid for all possible choices for repairing actions for the other missing is-a relations. Before computing the Source and Target sets, we introduce two new

[3] So this means that repairing one is-a relation may influence the repairing actions for other missing is-a relations. However, when *generating* repairing actions in the algorithm in Figure 4.5 the only influence that is taken into consideration is the fact that missing is-a relations are or will be repaired (least informative repairing action), but not the actual (possibly more informative) repairing actions that will be performed.

concepts x and y for each missing is-a relation (a,b) in the knowledge base as well as the axioms $a \rightarrow x$, $x \rightarrow y$, $y \rightarrow b$. (x,y) satisfies the requirements that each possible repairing action for (a,b) should satisfy. As they are new concepts in the knowledge base, the properties and relations of x, respectively y, to other concepts in the knowledge base represent the properties and relations that are common to the Source concepts, respectively Target concepts, of the possible repairing actions for (a,b). The Source and Target sets are now computed relative to the x and y.

Consider the case presented in Figure 4.8, where $\mathrm{O} = (\mathrm{C}, \mathrm{I})$ is an ontology with concepts $\mathrm{C} = \{1,2,3,4,5,6,7,8,9,10\}$ and is-a relations (shown in full lines in Figure 4.8) $\mathrm{I} = \{(7,6),(6,5),(5,2),(2,1),(7,4),(10,4),(10,9),(9,8),(8,3),(3,1),(4,1)\}$. (As before, I represents the is-a hierarchy and thus also all is-a relations derived from the elements in I.) The set of missing is-a relations (shown in dashed lines in Figure 4.8) is $\mathrm{M} = \{(5,4),(8,4)\}$.

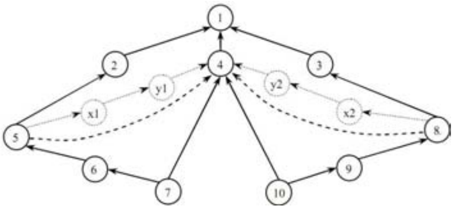

Figure 4.8: Example 2.

The basic algorithm in Figure 4.5 generates the following Source and Target sets.

$Source(5,4) = \{5,4,1,2\} - \{4,1\} = \{5,2\}$
$Target(5,4) = \{4,8,9,10,5,6,7\} - \{5,6,7\} = \{4,8,9,10\}$
$Source(8,4) = \{8,4,1,3\} - \{4,1\} = \{8,3\}$
$Target(8,4) = \{4,8,9,10,5,6,7\} - \{8,9,10\} = \{4,5,6,7\}$

The extended algorithm in Figure 4.7 adds the concepts x_1, y_1, x_2, y_2 and the is-a relations $5 \rightarrow x_1$, $x_1 \rightarrow y_1$, $y_1 \rightarrow 4$, $8 \rightarrow x_2$, $x_2 \rightarrow y_2$ and $y_2 \rightarrow 4$ (shown in dotted lines in Figure 4.8) in the knowledge base. It then generates the following Source and Target sets.

$Source\text{-}ext(5,4) = \{5,4,1,2,x_1,y_1\} - \{4,1,x_1,y_1\} = \{5,2\}$
$Target\text{-}ext(5,4) = \{4,8,9,10,5,6,7,x_1,y_1,x_2,y_2\} - \{5,6,7,x_1,y_1\}$
$= \{4,8,9,10,x_2,y_2\}$
$Source\text{-}ext(8,4) = \{8,4,1,3,x_2,y_2\} - \{4,1,x_2,y_2\} = \{8,3\}$
$Target\text{-}ext(8,4) = \{4,8,9,10,5,6,7,x_1,y_1,x_2,y_2\} - \{8,9,10,x_2,y_2\}$
$= \{4,5,6,7,x_1,y_1\}$

The sets generated by the extended algorithm indicate that there is an influence between the two missing is-a relations. Indeed, when a choice is made for repairing the first missing is-a relation, we have essentially added equivalence relations between x_1, respectively y_1, and concepts in the ontology. The appearance of x_1 and y_1 in the Target-ext set for the second missing is-a relation indicates that the concept chosen to be equivalent to x_1 (and all concepts between this concept and 5) are now also candidates for the Target-ext for the second missing is-a relation. For example, when choosing $(2,4)$ as a repairing action for missing is-a relation $(5,4)$ then $(3,2)$ is a possible repairing action for missing is-a relation $(8,4)$.

Similarly to the basic algorithm, the proposed repairing actions for a missing is-a relation (a,b) all lead to the derivation of (a,b) in the extended ontology. In general, a user may repair the ontology by choosing for each missing is-a relation (a,b) an element from $Source\text{-}ext(a,b)$ and an element from $Target\text{-}ext(a,b)$. However, as the algorithm only takes into account influences that are common to all possible choices for repairing actions, a user may want to repair one missing is-a relation and recompute repairing actions for the other missing is-a relations.

4.5.2 Ranking repairing actions

In general, there may be many missing is-a relations that need to be repaired. Although it is possible to repair the missing is-a relations in any order, it may be easier for the user to start with the ones where there are the fewest choices. Therefore, our ranking algorithm ranks the missing is-a relations according to the number of their possible repairing actions. For a missing is-a relation, it is calculated as the product of the Source set size and Target set size in case of the basic algorithm. For the extended algorithm, it is calculated in the same manner but without counting the extra-added new concepts.

4.5.3 Recommending repairing actions

For a missing is-a relation under repair, there may be too many possible repairing actions to choose from. Therefore, we developed an algorithm to recommend the most informative repairing actions based on some external domain knowledge. We assume that there is domain knowledge which we can query regarding subsumption between concepts. There are several such sources such as general thesauri (e.g. WordNet) or specialized domain-specific sources (e.g. the Unified Medical Language System).

In our algorithm (see Figure 4.9) we generate recommended repairing actions for a missing is-a relation starting from the Source and Target sets generated by the algorithm in Figure 4.5[4]. The algorithm selects the most informative repairing actions that are supported by evidence in the domain knowledge. The variable *visited* keeps track of already processed repairing actions. The variable *recommended* stores recommended repairing actions at each step and its final value is returned as output. It is initialized with the repairing action from the missing is-a relation itself. This is the least informative repairing action which is ensured to be correct. Steps 3 and 4 compute the set X_e of maximal elements with respect to the is-a relation in the Source set and the set Y_e of minimal elements with respect to the is-a relation in the Target set. The elements from $X_e \times Y_e$ are then the most informative repairing

[4]We have also extended the algorithm in Figure 4.9 to deal with Source-ext and Target-ext sets derived by the algorithm in Figure 4.7.

Input:
The ontology O and the missing is-a relation (a,b) under repair,
$Source(a,b)$ and $Target(a,b)$ computed by the basic algorithm.
Output
Recommended repairing actions.
Algorithm
Global Variable $visited$: stores already processed repairing actions.
Global Variable $recommended$: stores recommended repairing actions.
1. Set $visited = \{(a,b)\}$;
2. Set $recommended = \{(a,b)\}$;
3. Set $X_e = \{x_e : x_e \in Source(a,b) \bigwedge \forall x \in Source(a,b)$: if $x_e \rightarrow x$ then $x = x_e\}$;
4. Set $Y_e = \{y_e : y_e \in Target(a,b) \bigwedge \forall\, y \in Target(a,b)$: if $y \rightarrow y_e$ then $y = y_e\}$;
5. For each pair $(x_e, y_e) \in X_e \times Y_e$: call `FindRec` (x_e, y_e);
6. Return $recommended$;
Function `FindRec(concept x, concept y)`
i. If $(x,y) \in visited$ then return;
ii. Add (x,y) to $visited$;
iii. If $\exists\ (x_r, y_r) \in recommended$: $x \rightarrow x_r \wedge y_r \rightarrow y$ then return;
iv. If x is a sub-concept of y according to the domain knowledge, then
  Remove all (x_r, y_r) from $recommended$ for which $x_r \rightarrow x$ and $y \rightarrow y_r$;
  Add (x,y) to $recommended$;
 else
  Let Y_{sup} be the set of direct super-concepts of y;
  For each $y_s \in Y_{sup} \cap Target(a,b)$: call `FindRec` (x, y_s);
  Let X_{sub} be the set of direct sub-concepts of x;
  For each $x_s \in X_{sub} \cap Source(a,b)$: call `FindRec` (x_s, y);

Figure 4.9: Algorithm for recommending repairing actions.

actions. For each of these elements (x,y) we check whether there is support in the domain knowledge in step 5. Steps i and ii in the function `FindRec` do bookkeeping regarding the already processed repairing actions. Step iii assures that we do not recommend is-a relations that are less informative than others already recommended. In step iv we check whether there is support in the domain knowledge for the repairing action. If so, then the repairing action is recommended and all less informative repairing actions are removed from the recommendation set. If not, then we check whether there is support in the domain knowledge for the repairing actions that are less informative than (x,y). Among these we start with the most informative repairing actions.

4.5.4 Executing repairing actions

When a user has chosen and executed a repairing action for a particular missing is-a relation, it may influence the set of possible repairing actions for other missing is-a relations. Therefore, the repairing actions for the other missing is-a relations need to be recomputed based on the ontology extended with the chosen repairing action.

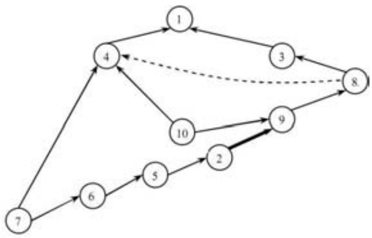

Figure 4.10: Example 2 - update.

For instance, Figure 4.10 shows the new situation when choosing the repairing action $(2,9)$ (shown in thick line) for missing is-a relation $(5,4)$ in the example in Figure 4.8. In this case the Source and Target sets become the following for the basic algorithm:

$Source(8,4) = \{8,4,1,3\} - \{4,1\} = \{8,3\}$
$Target(8,4) = \{4,8,9,10,2,5,6,7\} - \{8,9,10,2,5,6,7\} = \{4\}$
and the following for the extended algorithm:
$Source\text{-}ext(8,4) = \{8,4,1,3,x_2,y_2\} - \{4,1,x_2,y_2\} = \{8,3\}$
$Target\text{-}ext(8,4) = \{4,8,9,10,2,5,6,7,x_2,y_2\} - \{8,9,10,2,5,6,7,x_2,y_2\} = \{4\}$
When we compare the computed repairing actions after the choice of $(2,9)$ for repairing $(5,4)$ with the repairing actions computed before the choice (see section 4.5.1), we note that the repairing actions that introduce equivalence relations (e.g. $(8,6)$) are removed after the choice of $(2,9)$ (preference $\ll_{SH}$ in definition 4.2.13). However, before $(2,9)$ is chosen these repairing actions do not necessarily introduce equivalence relations. For instance, we could have repaired $(8,4)$ first using one of these repairing actions, and afterwards repaired $(5,4)$.

For small ontologies, computing the repairing actions does not take much time and the approach is feasible in a real setting. For large ontologies the computation time may not be small enough to guarantee immediate updates in an implemented tool for repairing. Therefore, in the algorithm[5] in Figure 4.11 we introduced a way to keep track of the influences between different missing is-a relations. The missing is-a relations for which the Source or Target sets can change are the missing is-a relations for which at least one of the concepts is a sub-concept or super-concept of at least one of the concepts in the chosen repairing action for the repaired missing is-a relation. We only update the Source and Target sets for these missing is-a relations. In addition, we also remove the other missing is-a relations that have been repaired by the current repairing action.

[5]Like the algorithm in Figure 4.9, this algorithm is applicable for cases using the basic algorithm. We also have a version with similar strategy for when we use the extended algorithm.

Input
The ontology under repair O, the repaired missing is-a relation (a_r, b_r), the repairing action (x_r, y_r) taken for (a_r, b_r), the set of non-repaired missing relations M_r.
Output
Updated Source and Target sets.
Algorithm
1. Add (x_r, y_r) to the KB;
2. For each missing is-a relation $(a, b) \in M_r$:
   If $a \rightarrow x_r$ then recompute super-concepts(a);
   If $b \rightarrow x_r$ then recompute super-concepts(b);
   If $a \rightarrow x_r$ or $b \rightarrow x_r$ then $Source(a,b)$:= super-concepts(a) $-$ super-concepts(b);
   If $y_r \rightarrow a$ then recompute sub-concepts(a);
   If $y_r \rightarrow b$ then recompute sub-concepts(b);
   If $y_r \rightarrow a$ or $y_r \rightarrow b$ then $Target(a,b)$:= sub-concepts(b) $-$ sub-concepts(a);

Figure 4.11: Algorithm for updating Source and Target sets.

4.6 Implemented System

We implemented a system RepOSE (*Rep*air of *O*ntological *S*tructure *E*nvironment) in Java based on our approach described in the previous section. We use a framework and reasoner provided by Jena[6] (version 2.5.7). Here, we show its use using pieces of MA and NCI-A regarding the concept *joint*, as well as a PRA with 8 equivalence mappings.

As input our system takes a set of ontologies in OWL format as well as a set of PRAs in RDF format. The ontologies and PRAs can be imported using the *Load Ontologies and PRAs* button. The user can see the list of ontologies in the *Ontologies* menu (see Figure 4.12). Once the *Detect Missing IS-A Relations* button is clicked, missing is-a relations are detected in all ontologies. Then, the user can select an ontology to repair, and the *Missing IS-A Relations* menu shows the missing is-a relations of the currently selected ontology. In this case the ontology *joint_mouse_anatomy.owl* is selected and it contains 7 missing is-a relations (same as the case in Figure 1).

Clicking on the *Generate Repairing Actions* button, results in the computation of repairing actions for the missing is-a relations of the ontology under repair, which is preceded by a two-stage preprocessing step. During the preprocessing, one stage is to identify the missing is-a relations which are actually equivalence relations and repair them by adding the equivalence relations. The other is to identify and remove the redundant missing is-a relations which are derivable from the ontology extended with other missing is-a relations. Then, repairing actions for each missing is-a relation are computed and presented as Source and Target sets. The selection of the *useExtendedAlg* checkbox makes the computation use our extended algorithm, otherwise our basic algorithm is used.

Once the Source and Target sets are computed, the missing is-a relations are

[6] http://jena.sourceforge.net/

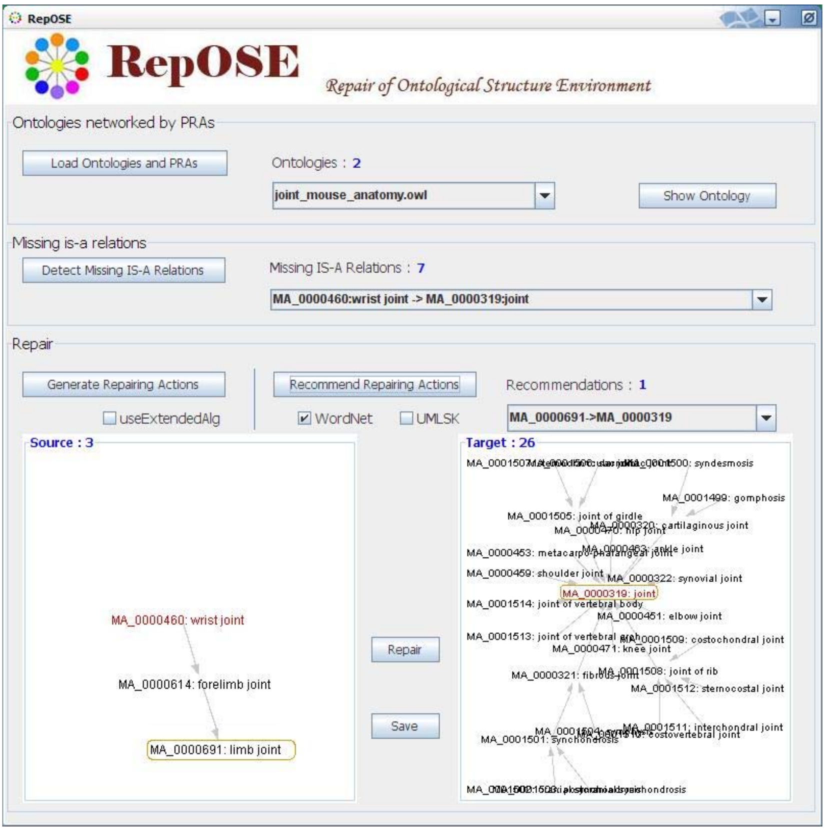

Figure 4.12: User interface of RepOSE.

ranked with respect to the number of possible repairing actions. The first missing is-a relation in the list has the fewest possible repairing actions, and may therefore be a good starting point. When the user chooses a missing is-a relation, the Source and Target sets for the repairing actions are shown in the panels on the left and the right, respectively (as shown in Figure 4.12). Both these panels have zoom control and could be opened in a separate window by double clicking. The concepts in the missing is-a relation are highlighted in red. In this case, the repairing actions of the missing is-a relations are generated using the basic algorithm. The selection of the missing is-a relation "*wrist joint* is-a *joint*" displays its Source and Target sets in the panels. They contain 3 and 26 concepts respectively.

For the selected missing is-a relation, the user can also ask for recommended repairing actions by clicking the *Recommend Repairing Actions* button. For the query of domain knowledge, we currently implemented two methods. The first method is based on WordNet, making use of the WordNet senses and hypernym relations to verify the subsumption relation between concepts. The second one is based on UMLS Knowledge Source Server, checking whether one concept is defined as an ancestor of another in UMLS Metathesaurus. On the interface, the two checkboxes allow the user to specify the external domain knowledge used for generating recommendations. In our case, the system uses *WordNet* and recommends to add an is-a relation between *limb joint* and *joint*. In general, the system presents a list of recommendations. By selecting an element in the list, the concepts in the recommended repairing action are identified by round boxes in the panels. The user can repair the missing is-a relation by selecting a concept in the Source panel and a concept in the Target panel and clicking on the *Repair* button. The repairing action is then added to the ontology, and other missing is-a relations are updated, as well as the set of missing is-a relations of every ontology in the network.

At all times during the process the user can inspect the ontology under repair by clicking the *Show Ontology* button. The is-a structure of the repaired ontology will be shown in a separate window with newly added is-a relations being highlighted. The user can save the repaired ontology into an OWL file by clicking the *Save* button, or select another ontology to repair. At all times the user can also switch to another ontology in the network. The whole debugging process runs semi-automatically until no more missing is-a relations are found or unrepaired in the network.

4.7 Experiment

In this section, we present an experiment using our debugging approach, where we debug a network consisting of the two ontologies and the PRA from the 2008 Anatomy track in OAEI. As described before, the two ontologies, MA and NCI-A contain 2744 and 3304 concepts respectively, while the PRA between them contains 988 equivalence mappings. The experiment was performed on an AMD Dual Core Processor 2.90GHZ desktop machine with 4 GB DDR2 memory under Windows Vista Business operating system (SP2) and Java 1.6 compiler.

4.7.1 Detecting missing is-a relations for the first time

After loading the two ontologies and the PRA, the running time of our detection algorithm is about 2 minutes. As a result, we found 199 missing is-a relations in MA and 167 in NCI-A. As discussed before, all these missing is-a relations are between PRA concepts. A further check shows that 6 missing is-a relations in MA and 3 in NCI-A are actually equivalence relations, and 78 in MA and 84 in NCI-A are redundant. When all these missing is-a relations are preprocessed before generating repairing actions, we have 115 missing is-a relations in MA and 80 in NCI-A.

4.7.2 Generating repairing actions for the first time

Results.

After preprocessing the missing is-a relations, we generate repairing actions for the remaining missing is-a relations in both ontologies. For MA, our basic algorithm generates for 9 missing is-a relations only 1 repairing action (which is then the missing is-a relation itself). This means that these could be immediately repaired. For NCI-A this number is 5. Of the remaining missing is-a relations there are 61 missing is-a relations for MA that have only 1 element in the Source set and 2 missing is-relations that have 1 element in the Target set. For NCI-A these numbers are 20 and 3, respectively. These are likely to be good starting points for repairing. Figures 4.13 shows for different ranges how many Source and Targets sets had a size in that range. We see that for most of the missing is-a relations these sets are small and thus can be easily visualized in the panels of our system.

	total	1	2-10	11-20	21-30	31-40	41-50	51-100	101-200	201-300	301-400	>400
MA - Source	115	70	45	0	0	0	0	0	0	0	0	0
MA - Target	115	11	50	5	9	4	6	5	18	3	0	4
NCI-A - Source	80	25	55	0	0	0	0	0	0	0	0	0
NCI-A - Target	80	8	52	6	2	0	0	5	4	1	2	0

Figure 4.13: Sizes of Source and Target sets.

Influences between missing is-a relations

Figure 4.14 shows the influences between different missing is-a relations that can be computed using our extended algorithm. In Figure 4.14 the last column (ST) shows the number of missing is-a relations where x's and y's of other missing is-a relations occur in both Source and Target sets. For the other columns the x's and y's only occur in Source or Target, but not in both. For instance, for MA there are 22 missing is-a relations whose Source or Target set contain x and y from one other missing is-a relation. We see that for a majority of the missing is-a relations (94/115 for MA and 67/80 for NCI-A) there are influences. An interesting observation is that in several cases missing is-a relations that have the same number of influences from other missing is-a relations, actually influence each other. For instance, in NCI-A we find missing is-a relations between each of *Bronchus_Basement_Membrane*, *Bronchus_Cartilage*, *Bronchus_Lamina_Propria*, *Bronchus_Submucosa*, and the concept *Bronchus_Connective_Tissue*. Repairing one of these missing is-a relations influences the repairing actions of all the others. We found several such clusters, among others for instance, in MA concerning *body cavity/lining*, *lymphoid tissue*, and *brain nucleus* with 7, 4 and 6 missing is-a relations, respectively.

	total	1	2	3	4	5	6	7	8	9	10	11-15	16-35	ST
MA	94	22	5	3	5	19	10	8	0	4	0	14	0	4
NCI-A	67	14	22	3	1	2	0	0	0	0	0	6	6	13

Figure 4.14: Influence between repairing actions of different missing is-a relations - in Source or Target.

4.7.3 Recommending repairing actions

We tested generating recommendations for all the missing is-a relations in both ontologies using WordNet only. The running time was circa 4 minutes for MA and circa 2 minutes for NCI-A. For NCI-A the system recommended[7] repairing actions for only 5 missing is-a relations and each of those received one recommended repairing action. For MA 19 missing is-a relations received 1 recommended repairing action, 12 received 2 and 2 received 3. The recommendation can come from small sets of repairing actions or from large sets. For instance, for MA the system recommends for the missing is-a relation *(mandible, bone)* the following three repairing actions *(oral region cartilage/bone, bone)*, *(viscerocranium bone, bone)*, and *(mandible, lower jaw)*. The repairing actions are recommended from a Source set of 177 concepts (and 15 influencing missing is-a relations) and a Target set of 3 concepts.

4.7.4 Executing repairing actions in a complete session

General comments on the complete debugging session

To obtain information on executing repairing actions, we have run a complete session using the basic algorithm for generating repairing actions. This test run was done by the authors. As we are not domain experts, we have used [104] to decide on possible choices and used the recommendation algorithm, although we cannot guarantee the correctness of all our repairs. However, this run already gives us some interesting information. It took about 3 hours to repair these two ontologies. In most cases the ranking and recommendations seemed useful. For 27 missing is-a relations in MA and 10 in NCI-A, the Target set was too large to have a good visualization in the tool.

[7] We do not count the missing is-a relation itself as a recommendation.

Influence of repairing actions on repairing actions for other missing is-a relations

Repairing influenced the number of repairing actions for other missing is-a relations. During the repairing process, for 25 missing is-a relations in MA and 11 in NCI-A the number of repairing actions changed. For each of these missing is-a relations the size of the Target sets increased while the Source sets remained the same. The increase of the number of possible repairing actions ranged from 1 to 35 with average 8 in MA, and from 4 to 24 with average 12 in NCI-A.

Influence of repairing actions on the set of missing is-a relations

Repairing also influenced the set of missing is-a relations. In MA 19 missing is-a relations were repaired due to the repairing of other missing is-a relations.

Detection of new missing is-a relations

We found 6 new missing is-a relations in MA and 10 in NCI-A. Every newly derived missing is-a relation was caused by the repairing of a missing is-a relation in the other ontology. New missing is-a relations appeared only when the relevant repaired missing is-a relation used a repairing action other than itself. This is as it should be, as using a missing is-a relation as repairing action for itself, does not change the induced ontology for the ontology network and thus no new missing is-a relations would be derived from the ontology network.

Among these new missing is-a relations, 4 in MA and 4 in NCI-A appear separately after the execution of the relevant repairing action. Common to these new missing is-a relations is that each concept in the new missing is-a relation in one ontology is equivalent to a concept in the executed repairing action in the other ontology. For instance, the new missing is-a relation *(Thoracic_Aorta, Artery)* in NCI-A was caused by the execution of repairing action *(thoracic aorta, artery)* in MA, and the new missing is-a relation *(venule endothelium, vein endothelium)* in MA was caused by the execution of *(Venule_Endothelium, Vein_Endothelium)* in NCI-A. Other new missing is-a relations appeared together in the respective ontology, of which 1 and 3 were redundant in MA and NCI-A, respectively. For example, in NCI-A, after the missing is-a relation *(Myocardium, Muscle)* in MA was repaired by *(Striated_Muscle_Tissue, Muscle)*, the new missing is-a relations *(skeletal muscle tissue, muscle)* and *(striated muscle tissue, muscle)* appeared together, in which the former was redundant with respect to the latter.

4.8 Related Work

4.8.1 Ontology Debugging

Debugging Modeling Defects

There is not much work on debugging modeling defects in networked ontologies. The work closest to our own is [105], in which the authors deal with the missing is-a relations as ontology nonalignments in the context of ontology enrichment. Given two pairs of terms between two ontologies which are linked by the same kind of relationship, if the two terms in one ontology are linked by an is-a relation while the corresponding terms in the other are not, it is deemed as a nonalignment. However, the authors note that, depending on the specific relationship, not all nonalignments are defects and there is no conclusive solution for repairing.

The problem of modeling defects is also addressed in [106], where ontology repair is used when a formula can be derived from an ontology, but, in the words of the authors, it is not correct according to the world. In this case a mapping is computed such that the mapped formula is correct according to the world and can be derived from the mapped ontology, or such that the mapped formula cannot be derived from the mapped ontology. The setting where this is used is a framework where agents use ontologies and when certain tasks cannot be performed, communication between the agents takes place to identify mismatches between the ontologies and revise the ontologies.

In [14] we discussed the use of a PRA in the setting of ontology alignment. One of the approaches included detecting missing is-a relations by using the structure of the ontologies and the PRA. Missing is-a relations were found by looking at pairs of equivalence mappings. If there is an is-a relation between the terms in the mappings belonging to one ontology, but there is no is-a relation between the corresponding terms in the other ontology, then we concluded that an is-a relation is missing in the second ontology. The detected missing is-a relations were then added to the ontologies. This is the simplest kind of structural repair.

In [16], we focused on repairing a given set of missing is-a relations in a *single* ontology. The work contained algorithms for generating, recommending and executing repairing actions. However, there was no investigation on how to detect missing is-a relations using networked ontologies and no attempt to find new missing is-a relations during the repairing.

Debugging Semantic Defects

There is more work that addresses semantic defects in ontologies. Most of them focus on single and isolated ontologies, and there is few support for ontologies connected by mappings [74].

In [107] minimal sets of axioms are identified which need to be removed to turn an ontology coherent. In [72, 108, 109] strategies are described for repairing unsatisfiable concepts detected by reasoners, explanation of errors, ranking erroneous axioms, and generating repair plans. In [110] the focus is on maintaining the consistency as the ontology evolves. It formalized the semantics of change for the OWL ontologies and proposed methods for detecting and resolving inconsistency at three different levels.

In [111] and [112] the setting is extended to repairing ontologies connected by mappings. In this case, semantic defects may be introduced by integrating ontologies. Both works assume that ontologies are more reliable than the mappings and try to remove some of the mappings to restore consistency. The solutions are often based on the computation of minimal unsatisfiability-preserving sets or minimal conflict sets. The work in [113] further characterizes the problem as mapping revision. Using the theory of belief base revision [114], it gives a rationality analysis for the logical properties of the revision algorithms. In contrast, the work in [115] not only deals with the inconsistencies introduced by the integration of ontologies, but also unintended entailments confirmed by the user. Depending on the user's judgement on the new entailments, the repairing could be done on both the ontologies and the mappings by removing a minimal set of axioms.

4.8.2 Relation discovery in ontologies

Related to the detection of missing relations, there is much work on finding relationships between terms in the ontology learning area [116]. In this setting, new ontology elements are derived from structured data (such as databases), semi-structured data or unstructured data (e.g., text documents) using well-established techniques from a variety of disciplines [117].

Regarding the discovery of subsumption relations, one paradigm is based on linguistics using lexico-syntactic patterns. The pioneering research conducted in this line is in [118], which defines a set of patterns indicating is-a relationships between words in the text. However, depending on the chosen corpora, these patterns may occur rarely. Thus, though the approach has a reasonable precision, its recall is very low. To overcome this, an approach proposed in [119] collects a corpus from the WWW using Google and identifies patterns from this collective knowledge. An evaluation and extension of this work is in [120]. Another work in this category is done on the OBO Foundry ontologies [121] [122], where the authors make use of the nomenclature patterns of the biomedical terms within ontologies to derive computable logical definitions, whereby the reasoner can be further applied to detect new relationships in these ontologies. In [123], two approaches for automatic prediction of candidate terms for GO are studied, in which one approach, superstring prediction, uses the compositional pattern of GO terms to find new subconcepts.

Another paradigm is based on machine learning and statistical methods, such as k-nearest neighbors approach [124], bottom-up hierarchical clustering techniques [125], supervised classification [126] and formal concept analysis [127].

Compared with these approaches, our detection method uses the ontology network as the domain knowledge for the discovery of is-a relations. It is able to deal with multiple ontologies at the same time rather than a single ontology. However, these approaches are complementary to our detection method, in that, results from them, after validation, could be used as supplement to the domain knowledge.

4.8.3 Belief Revision

From the perspective of belief revision [128] [129], our repairing approach could be deemed as an instantiation of belief expansion. It accommodates the structural repair as a new piece of information, and adds it to the ontology without checking consistency. Since in our setting the ontologies contain only named concepts and subsumption axioms, our repairing approach does not introduce logical contradictions.

4.9 Summary

In this chapter we have focused on one of the important kinds of defects in ontologies, namely the modeling defects, and in particular, defects in the is-a structure of ontologies. We have proposed an approach for debugging the missing is-a structure of ontologies that are networked by a set of PRAs. We defined important notions and developed algorithms for detection and repair of missing is-a structure of ontologies. We also implemented a system and showed an experiment using two well-known ontologies.

Chapter 5

Conclusion and Future Work

This chapter concludes the work in the thesis and discusses the possible directions for the future work.

5.1 Conclusion

With the goal of improving the quality of ontology-based semantically-enabled applications, the work presented in this thesis contributes to dealing with missing mappings and structure in a network of ontologies.

As for the problem of missing mappings between networked ontologies, we have proposed an ontology alignment methodology, which uses the existing mappings as PRAs to find more mappings between ontologies. We have done an experimental study on the use of the PRA in the preprocessing step for partitioning the ontologies, the use of PRA in the matcher for changing the similarity values of mapping suggestions and the use of PRA for filtering mapping suggestions. We have also investigated the influence of the size of the PRA on the quality of the alignment results.

For the problem of missing structure of networked ontologies, we have presented an ontology debugging approach, which considers the missing is-a structure as a kind of modeling defects. We have proposed methods of finding the missing is-a relations using the ontology network as the domain knowledge, and methods of generating and recommending possible ways of repairing and executing the repairing. We also introduced an implemented system RepOSE, which allows a domain expert to debug the missing is-a structure of ontologies in a semi-automatic way.

5.2 Future Work

There are a number of directions that are interesting for the future work.

Regarding the missing mappings between networked ontologies, we would like to test the alignment algorithms using PRAs on other ontologies and with different base algorithms. Combinations and interactions of the methods should be investigated. It would also be interesting to look at other kinds of patterns in the alignment data. Further, since we only consider the mappings with 1-1 equivalence relations, our approaches need to be extended for other kinds of mappings (e.g. is-a mappings). Finally, the approach should be integrated in an iterative ontology alignment framework, where we need to handle the situation that the PRA is continuously growing involving both the accepted and the rejected mapping suggestions.

As for the missing structure in networked ontologies, since our current approach uses PRAs as domain knowledge assuming that the given mappings are correct, a direct extension would be the case when these mappings are not necessarily correct (e.g. they are generated by an ontology alignment system and not yet validated). In this case, we will need to also deal with the repairing of the mappings. Another interesting direction is to deal with ontologies represented in more expressive representation languages, and investigate possible influences between semantic defects and modeling effects. We also want to find ways to optimize the generation of results. For instance, as the generation of structural repairs may take a lot of time, we may want to investigate ways to partition the set of missing is-a relations into parts that can be processed independently.

Another direction we want to explore is the interaction and integration of ontology alignment and ontology debugging, since current work has demonstrated that, mappings between ontologies could be used for detecting and repairing missing is-a structure of networked ontologies, while ontologies with repaired structure could help to find more correct mappings. We will investigate possible ways of interleaving ontology alignment and ontology debugging steps, as well as specialized algorithms to compute the consequences of updated mappings and is-a relations of ontologies.

Bibliography

[1] T Berners-Lee and M Fischetti. *Weaving the Web: The Past, Present and Future of the World Wide Web by its Inventor*. Texere, 2000.

[2] MR Koivunen and E Miller. W3C semantic web activity. *Semantic Web Kick-Off in Finland*, pages 27–44, 2001.

[3] MC Daconta, LJ Obrst, and KT Smith. *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Wiley, 2003.

[4] D Trastour, C Bartolini, and C Preist. Semantic Web Support for the Business-to-Business E-Commerce Lifecycle. In *Proceedings of the 11th International Conference on World Wide Web*, pages 89–98. ACM, 2002.

[5] P Lord, S Bechhofer, MD Wilkinson, G Schiltz, D Gessler, D Hull, CA Goble, and L Stein. Applying Semantic Web Services to Bioinformatics Experiences Gained, Lessons Learnt. In *Proceedings of the 3rd International Semantic Web Conference*, volume 3298 of *LNCS*, pages 350–364. Springer-Verlag, 2004.

[6] R Neches, R Fikes, T Finin, T Gruber, R Patil, T Senator, and W Swartout. Enabling Technology for Knowledge Sharing. *AI Magazine*, 12(3):36–56, 1991.

[7] P Lambrix, H Tan, V Jakonienė, and L Strömbäck. Biological Ontologies. In Baker and Cheung, editors, *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, chapter 4, pages 85–99. Springer, 2007.

[8] N Shadbolt, T Berners-Lee, and W Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, July 2006.

[9] P Shvaiko and J Euzenat. Ten challenges for ontology matching. In *Proceedings of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics*, volume 5332 of *LNCS*, pages 1164–1182, 2008.

[10] P Lambrix, L Strömbäck, and H Tan. Information Integration in Bioinformatics with Ontologies and Standards. In Bry and Maluszynski, editors, *Semantic Techniques for the Web: The REWERSE perspective*, chapter 8, pages 343–376. Springer, 2009.

[11] C Conroy, R Brennan, D O'Sullivan, and D Lewis. User Evaluation Study of a Tagging Approach to Semantic Mapping. In *Proceedings of the 6th European Semantic Web Conference*, LNCS, pages 623–637, 2009.

[12] J Euzenat, A Ferrara, L Hollink, A Isaac, C Joslyn, V Malaisé, C Meilicke, A Nikolov, J Pane, M Sabou, F Scharffe, P Shvaiko, V Spiliopoulos, H Stuckenschmidt, O Šváb-Zamazal, V Svátek, C Trojahn, G Vouros, and S Wang. Results of the Ontology Alignment Evaluation Initiative 2009. In P Shvaiko, J Euzenat, F Giunchiglia, H Stuckenschmidt, N Noy, and A Rosenthal, editors, *Proceedings of the 4th International Workshop on Ontology Matching*, pages 73–126, 2009.

[13] P Lambrix and H Tan. SAMBO–A System for Aligning and Merging Biomedical Ontologies. *Journal of Web Semantics, Special issue on Semantic Web for the Life Sciences*, 4(3):196–206, 2006.

[14] P Lambrix and Q Liu. Using partial reference alignments to align ontologies. In *Proceedings of the 6th International Workshop on Data Integration in the Life Sciences*, volume 5554 of *LNCS*, pages 188–202. Springer-Verlag, 2009.

[15] P Lambrix and Q Liu. Debugging the Missing is-a Structure within Ontologies Networked by Partial Reference Alignments. submitted, 2010.

[16] P Lambrix, Q Liu, and H Tan. Repairing the Missing is-a Structure of Ontologies. In *Proceedings of the 4th Asian Semantic Web Conference*, volume 5926 of *LNCS*, pages 76–90. Springer-Verlag, 2009.

[17] D Davis. Project10X's Semantic Wave 2008 Report: Industry Roadmap to Web 3.0 and Multibillion Dollar Market Opportunities. Technical report, 2008.

[18] N Guarino and P Giaretta. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N Mars, editor, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pages 25–32. IOS Press, 1995.

[19] TR Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 43:907–928, 1995.

[20] B Smith and C Welty. Ontology-Towards a New Synthesis. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, pages 3–9, 2002.

[21] A Gómez-Pérez. Ontological Engineering: A State Of The Art. In *Expert Update*, pages 33–43, 1999.

[22] R Stevens, CA Goble, and S Bechhofer. Ontology-based Knowledge Representation for Bioinformatics. *Briefings in Bioinformatics*, 1(4):398–414, 2000.

[23] ISO 2788. *Guidelines for the establishment and development of monolingual thesauri*. International Organization for Standarization, 1986.

[24] M Uschold and M Gruninger. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11:93–136, 1996.

[25] M Uschold and M Gruninger. A Framework for Understanding and Classifying Ontology Applications. In *Proceedings of the 12th Workshop on Knowledge Acquisition Modeling and Management*, 1999.

[26] F Manola, E Miller, and B McBride. RDF Primer. *W3C Recommendation*, 10, 2004.

[27] D Brickley, RV Guha, and B McBride. RDF Vocabulary Description Language 1.0: RDF Schema. *W3C Recommendation*, 10, 2004.

[28] J Heflin, J Hendler, and S Luke. SHOE: A knowledge representation language for internet applications. Technical report, Dept. of Computer Science, University of Maryland at College Par, 1999. Technical Report CS-TR-4078 (UMIACS TR-99-71).

[29] D Fensel, FV Harmelen, I Horrocks, DL McGuinness, and PF Patel-Schneider. OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.

[30] I Horrocks. DAML+OIL: a Description Logic for the Semantic Web. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 51(4):4–9, 2002.

[31] S Bechhofer, FV Harmelen, J Hendler, I Horrocks, DL McGuinness, PF Patel-Schneider, and LA Stein. OWL web ontology language reference. *W3C Recommendation*, 10, 2004.

[32] G Antoniou and F Harmelen. Web ontology language: Owl. *Handbook on ontologies*, pages 91–110, 2009.

[33] S Staab and R Studer. *Handbook on ontologies*. Springer Verlag, 2004.

[34] I Horrocks, PP Schneider, and FV Harmelen. From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1):7–26, 2003.

[35] B Chandrasekaran, JR Josephson, and VR Benjamins. What are ontologies, and why do we need them? *Intelligent Systems and Their Applications, IEEE*, 14(1):20–26, 2002.

[36] V Mascardi, V Cordì, and P Rosso. A Comparison of Upper Ontologies (Technical Report DISI-TR-06-21). Technical report, 2007.

[37] NF Noy. Semantic integration: a survey of ontology-based approaches. *SIGMOD Record*, 33(4):65–70, 2004. Special Issue on Semantic Integration.

[38] DB Lenat and RV Guha. *Building large knowledge-based systems: representation and inference in the Cyc project*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[39] J He, H Lai, and H Wang. A Cyc-Based Multi-agent System. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, page 63. IEEE Computer Society, 2008.

[40] J Curtis, J Cabral, and D Baxter. On the Application of the Cyc Ontology to Word Sense Disambiguation. In *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference*, pages 652–657, 2006.

[41] K Coursey. Living in CyN: Mating AIML and Cyc Together with Program N. *Resources avalaible at: http://www.daxtron.com*, 2004.

[42] C Fellbaum. *WordNet: An electronic lexical database*. The MIT press, 1998.

[43] MV Assem, A Gangemi, and G Schreiber. Conversion of WordNet to a standard RDF/OWL representation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 2006.

[44] A Pease, I Niles, and J Li. The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. In *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, 2002.

[45] I Niles and A Pease. Towards a standard upper ontology. In *Proceedings of the International Conference on Formal Ontology in Information System*, pages 2–9. ACM New York, NY, USA, 2001.

[46] C Masolo, S Borgo, A Gangemi, N Guarino, and A Oltramari. WonderWeb Deliverable D18, Ontology Library (final). Technical report, IST Project 2001-33052. WonderWeb: Ontology Infrastructure for the Semantic Web, 2009.

[47] P Grenon, B Smith, and L Goldberg. Biodynamic ontology: Applying BFO in the biomedical domain. In *Ontologies in Medicine - Proceedings of the Workshop on Medical Ontologies*, volume 201, pages 20–38. IOS Press, 2004.

[48] J Breuker, A Valente, and R Winkels. Legal ontologies in knowledge engineering and information management. *Artificial Intelligence and Law*, 12(4):241–277, 2004.

[49] MF Lopez, A Gómez-Pérez, JP Sierra, and AP Sierra. Building a Chemical Ontology using Methontology and the Ontology Design Environment. *Intelligent Systems and Their Applications, IEEE*, 14(1):37–46, 2002.

[50] A Frank. Spatial ontology: A geographical information point of view. *Spatial and Temporal Reasoning*, pages 135–153, 1997.

[51] T Athanasiadis, V Tzouvaras, K Petridis, F Precioso, Y Avrithi, and Y Kompatsiaris. Using a multimedia ontology infrastructure for semantic annotation of multimedia content. In *Proceedings of the 5th International Workshop on Knowledge Markup and Semantic Annotation at the 4th International Semantic Web Conference*, 2005.

[52] CB Jones, R Purves, A Ruas, M Sanderson, M Sester, MV Kreveld, and R Weibel. Spatial information retrieval and geographical ontologies an overview of the SPIRIT project. In *Proceedings of the 25th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 388. ACM, 2002.

[53] S Stephens, D LaVigna, M DiLascio, and J Luciano. Aggregation of bioinformatics data using Semantic Web technology. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(3):216–221, 2006.

[54] LF Soualmia, C Golbreich, and SJ Darmoni. Representing the MeSH in OWL: Towards a semiautomatic migration. In *Proceedings of the KR 2004 Workshop on Formal Biomedical Knowledge Representation*, pages 81–87, 2004.

[55] W Ceusters, B Smith, A Kumar, and C Dhaen. Ontology-based error detection in SNOMED-CT. In *Proceedings of MEDINFO*, pages 482–486, 2004.

[56] MQ Stearns, C Price, KA Spackman, and AY Wang. SNOMED clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium*, page 662. American Medical Informatics Association, 2001.

[57] M Ashburner, CA Ball, JA Blake, D Botstein, H Butler, JM Cherry, AP Davis, K Dolinski, SS Dwight, JT Eppig, MA Harris, DP Hill, L Issel-Tarver, A Kasarskis, S Lewis, JC Matese, JE Richardson, M Ringwald, GM Rubin, and G Sherlock. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25(1):25–29, 2000.

[58] E Camon, M Magrane, D Barrell, D Binns, W Fleischmann, P Kersey, N Mulder, T Oinn, J Maslen, A Cox, and R Apweiler. The gene ontology annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro. *Genome Research*, 13(4):662, 2003.

[59] C Rosse, A Kumar, JLV Mejino, DL Cook, LD Detwilern, and B Smith. A strategy for improving and integrating biomedical ontologies. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 639–643, 2005.

[60] E Beisswanger, S Schulz, H Stenzhorn, and U Hahn. BioTop: An upper domain ontology for the life sciencesA description of its current structure, contents and interfaces to OBO ontologies. *Applied Ontology*, 3(4):205–212, 2008.

[61] B Smith, W Ceusters, B Klagges, J Köhler, A Kumar, J Lomax, C Mungall, F Neuhaus, AL Rector, and C Rosse. Relations in biomedical ontologies. *Genome biology*, 6(5):R46, 2005.

[62] S Schulz, E Beisswanger, L Van Den Hoek, O Bodenreider, and EM Van Mulligen. Alignment of the UMLS semantic network with BioTop: methodology and assessment. *Bioinformatics*, 25(12):69–76, 2009.

[63] P Lambrix, H Tan, and Q Liu. SAMBO and SAMBOdtf results for the ontology alignment evaluation initiative 2008. In *Proceedings of the 3rd International Workshop on Ontology Matching*, volume 431 of *CEUR*, pages 190–198, 2008.

[64] P Shvaiko and J Euzenat. A Survey of Schema-based Matching Approaches. *Journal on Data Semantics*, IV:146–171, 2005.

[65] NF Noy and M Klein. Ontology Evolution: Not the Same as Schema Evolution. *Knowledge and Information Systems*, 6:428–440, 2004.

[66] Y Kalfoglou and M Schorlemmer. Ontology mapping: the state of the art. *Knowledge Engineering Review*, 18(1):1–31, 2003.

[67] J Euzenat and P Shvaiko. *Ontology Matching*. Springer, 2007.

[68] J Euzenat, TL Bach, J Barrasa, P Bouquet, JD Bo, R Dieng-Kuntz, M Ehrig, M Hauswirth, M Jarrar, R Lara, D Maynard, A Napoli, G Stamou, H Stuckenschmidt, P Shvaiko, S Tessaris, SV Acker, and I Zaihrayeu. State of the Art on Ontology Alignment. deliverable 2.2.3, Knowledge Web NoE, 2004.

[69] E Rahm and PA Bernstein. A Survey of Approaches to Automatic Schema Matching. *VLDB Journal*, 10:2001, 2001.

[70] H Tan, V Jakoniene, P Lambrix, J Aberg, and N Shahmehri. Alignment of Biomedical Ontologies using Life Science Literature. In *Proceedings of the International Workshop on Knowledge Discovery in Life Science Literature*, volume 3886 of *LNBI*, pages 1–17. Springer-Verlag, 2006.

[71] A Maedche and S Staab. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2), 2001. Special Issue on Semantic Web.

[72] A Kalyanpur, B Parsia, E Sirin, and J Hendler. Debugging Unsatisfiable Classes in OWL Ontologies. *Journal of Web Semantics*, 3(4):268–293, 2006.

[73] T Meyer, K Lee, R Booth, and JZ Pan. Finding Maximally Satisfiable Terminologies for the Description Logic ALC. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 269–274. AAAI Press, 2006.

[74] P Haase and G Qi. An Analysis of Approaches to Resolving Inconsistencies in DL-based Ontologies. In *Proceedings of International Workshop on Ontology Dynamics*, pages 188–202, 2007.

[75] V Haarslev and R Müller. RACER system description. *Automated Reasoning*, pages 701–705, 2001.

[76] B Parsia and E Sirin. Pellet: An OWL DL Reasoner. In V Haarslev and R Möller, editors, *Proceedins of the 2004 International Workshop on Description Logics*, volume 104 of *CEUR Workshop Proceedings*, 2004.

[77] I Horrocks. FaCT and iFaCT. In *Proceedings of the 1999 Description Logic Workshop*, volume 22, pages 133–135, 1999.

[78] G Flouris, D Manakanatas, H Kondylakis, D Plexousakis, and G Antoniou. Ontology Change: Classification and Survey. *Knowledge Engineering Review*, 23(2):117–152, 2008.

[79] NF Noy, N Griffith, and M Musen. Collecting Community-based Mappings in an Ontology Repository. In *Proceedings of the 7th International Semantic Web Conference*, volume 5318 of *LNCS*, pages 371–386, 2008.

[80] C Caraciolo, J Euzenat, L Hollink, R Ichise, A Isaac, V Malaisé, C Meilicke, J Pane, P Shvaiko, H Stuckenschmidt, O Svab, and V Svatek. Results of the Ontology Alignment Evaluation Initiative 2008. In P Shvaiko, J Euzenat, F Giunchiglia, and H Stuckenschmidt, editors, *Proceedings of the 3rd International Workshop on Ontology Matching*, pages 73–119, 2008.

[81] P Lambrix, H Tan, and W Xu. Literature-based Alignment of Ontologies. In *Proceedings of the 3rd International Workshop on Ontology Matching*, pages 219–223, 2008.

[82] B Chen, H Tan, and P Lambrix. Structure-based Filtering for Ontology Alignment. In *Proceedings of IEEE WETICE Workshop on Semantic Technologies in Collaborative Applications*, pages 364–369, 2006.

[83] T Takai-Igarashi, Y Nadaoka, and T Kaminuma. A Database for Cell Signaling Networks. *Journal of Computational Biology*, 5(4):747–754, 1998.

[84] M Ehrig and S Staab. QOM – Quick Ontology Mapping. In *Proceedings of the 3rd International Semantic Web Conference*, volume 3298 of *LNCS*, pages 683–697, 2004.

[85] J Li, J Tang, Y Li, and Q Luo. RiMOM: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1218–1232, 2009.

[86] X Zhang, Q Zhong, J Li, J Tang, G Xie, and H Li. RiMOM results for OAEI 2008. In *Proceedings of the 3rd International Workshop on Ontology Matching*, volume 431 of *CEUR*, page 182, 2008.

[87] YR Jean-Mary, EP Shironoshita, and MR Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):235–251, 2009.

[88] MH Seddiqui and M Aono. An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):344–356, 2009.

[89] MH Seddiqui and M Aono. Anchor-Flood: Results for OAEI-2009. In *Proceedings of the 4th International Workshop on Ontology Matching*, 2009.

[90] F Hamdi, B Safar, C Reynaud, and H Zargayouna. Alignment-based Partitioning of Large-scale Ontologies. *Advances in Knowledge Discovery and Management*, pages 251–269, 2010.

[91] W Hu, Y Zhao, and Y Qu. Partition-based block matching of large class hierarchies. In R Mizoguchi, Z Shi, and F Giunchiglia, editors, *Proceedings of the 1st Asian Semantic Web Conference*, volume 4185 of *LNCS*, pages 72–83. Springer-Verlag, 2006.

[92] F Hamdi, B Safar, N Niraula, and C Reynaud. TaxoMap in the OAEI 2009 alignment contest. In *Proceedings of the 4th International Workshop on Ontology Matching*, pages 230–237, 2009.

[93] IF Cruz, FP Antonelli, and C Stroe. AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment*, 2(2):1586–1589, 2009.

[94] IF Cruz, FP Antonelli, C Stroe, UC Keles, and A Maduko. Using AgreementMaker to Align Ontologies for OAEI 2009: Overview, Results, and Outlook. In *Proceedings of the 4th International Workshop on Ontology Matching*, LNCS, pages 135–146, 2009.

[95] J Noessner and M Niepert. CODI: Combinatorial Optimization for Data Integration–Results for OAEI 2010. In *Proceedings of the 5th International Workshop on Ontology Matching*, page 142, 2010.

[96] J Euzenat, A Ferrara, C Meilicke, J Pane, F Scharffe, P Shvaiko, H Stuckenschmidt, O Šváb-Zamazal, V Svátek, and C Trojahn. First results of the Ontology Alignment Evaluation Initiative 2010. In P Shvaiko, J Euzenat, F Giunchiglia, H Stuckenschmidt, M Mao, and I Cruz, editors, *Proceedings of the 5th International Workshop on Ontology Matching*, 2010.

[97] W Hu and Y Qu. Falcon-AO: A practical ontology matching system. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):237–239, 2007.

[98] NF Noy and MA Musen. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In *Workshop on Ontologies and Information Sharing at the 17th International Joint Conference on Artificial Intelligence*, pages 63–70, 2001.

[99] S Melnik, H Garcia-Molina, and E Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In *Proceedings of the 18th International Conference on Data Engineering*, ICDE, pages 117–128. IEEE Computer Society, 2002.

[100] J Madhavan, Ph Bernstein, and E Rahm. Generic Schema Matching with Cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases*, pages 49–58. Morgan Kaufmann Publishers Inc., 2001.

[101] F Giunchiglia, P Shvaiko, and M Yatskevich. Discovering missing background knowledge in ontology matching. In *Proceeding of the 17th European Conference on Artificial Intelligence*, pages 382–386. IOS Press, 2006.

[102] M Ehrig, S Staab, and Y Sure. Bootstrapping Ontology Alignment Methods with APFEL. In *Proceedings of the 4th International Semantic Web Conference*, volume 3729 of *LNCS*, pages 186–200. Springer-Verlag, 2005.

[103] H Tan and P Lambrix. A Method for Recommending Ontology Alignment Strategies. In *Proceedings of the 6th International Semantic Web Conference*, volume 4825 of *LNCS*, pages 494–507. Springer-Verlag, 2007.

[104] F Feneis and W Dauber. *Pocket Atlas of Human Anatomy*. Thieme Verlag, 4 edition, 2000.

[105] M Bada and L Hunter. Identification of OBO Nonalignments and Its Implications for OBO Enrichment. *Bioinformatics*, 24(12):1448–1455, 2008.

[106] F McNeill and A Bundy. Dynamic, Automatic, First-order Ontology Repair by Diagnosis of Failed Plan Execution. *International Journal on Semantic Web and Information Systems*, 3(3):1–35, 2007.

[107] S Schlobach. Debugging and Semantic Clarification by Pinpointing. In *Proceedings of the 2nd European Semantic Web Conference*, volume 3532 of *LNCS*, pages 226–240. Springer-Verlag, 2005.

[108] A Kalyanpur, B Parsia, E Sirin, and B Cuenca-Gray. Repairing Unsatisfiable Concepts in OWL Ontologies. In *Proceedings of the 3rd European Semantic Web Conference*, volume 4011 of *LNCS*, pages 170–184. Springer-Verlag, 2006.

[109] A Kalyanpur. *Debugging and Repair of OWL ontologies*. PhD thesis, University of Maryland at College Park, 2006.

[110] P Haase and L Stojanovic. Consistent Evolution of OWL Ontologies. In A Gómez-Pérez and J Euzenat, editors, *The Semantic Web: Research and Applications*, volume 3532 of *LNCS*, pages 182–197. Springer Berlin/Heidelberg, 2005.

[111] C Meilicke, H Stuckenschmidt, and A Tamilin. Repairing Ontology Mappings. In *Proceedings of the 22th Conference on Artificial Intelligence*, volume 2, pages 1408–1413. AAAI Press, 2007.

[112] Q Ji, P Haase, G Qi, P Hitzler, and S Stadtmuller. RaDON - repair and diagnosis in ontology networks. In *Proceedings of the 6th European Semantic Web Conference*, volume 5554 of *LNCS*, pages 863–867. Springer-Verlag, 2009.

[113] G Qi, Q Ji, and P Haase. A Conflict-Based Operator for Mapping Revision. In *Proceedings of the 8th International Semantic Web Conference*, volume 5823 of *LNCS*, pages 521–536. Springer-Verlag, 2009.

[114] SO Hansson. *A Textbook of Belief Dynamics Theory: Theory Change and Database Updating*. Kluwer Academic Publishers, 1999.

[115] E Jimenez-Ruiz, B Cuenca Grau, I Horrocks, and R Berlanga. Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences. In *Proceedings of the 6th European Semantic Web Conference*, volume 5554 of *LNCS*, pages 173–187. Springer-Verlag, 2009.

[116] Ph Cimiano, P Buitelaar, and B Magnini. *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press, 2005.

[117] Ph Cimiano, A Mädche, S Staab, and J Völker. Ontology Learning. *Handbook on Ontologies*, pages 245–267, 2009.

[118] MA Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, volume 2, pages 539–545, July 1992.

[119] Ph Cimiano and S Staab. Learning by googling. *ACM SIGKDD Explorations Newsletter*, 6(2):24–33, 2004.

[120] WR van Hage, S Katrenko, and G Schreiber. A Method to Combine Linguistic Ontology-Mapping Techniques. In *Proceedings of the 4th International Semantic Web Conference*, volume 3729 of *LNCS*, pages 732–744. Springer-Verlag, 2005.

[121] CJ Mungall. Obol: integrating language and meaning in bio-ontologies. *Comparative and Functional Genomics*, 5(6-7):509–520, 2004.

[122] CJ Mungall, M Bada, TZ Berardini, J Deegan, A Ireland, MA Harris, DP Hill, and J Lomax. Cross-product extensions of the Gene Ontology. *Journal of Biomedical Informatics*, (accepted), 2010.

[123] T Wächter, H Tan, A Wobst, P Lambrix, and M Schroeder. A corpus-driven approach for design, evolution and alignment of ontologies. In *Proceedings of the Winter Simulation Conference*, pages 1595–1602, 2006. Invited contribution.

[124] A Maedche, V Pekar, and S Staab. Ontology Learning Part One – On Discovering Taxonomic Relations from the Web. In Zhong, Liu, and Yao, editors, *Web Intelligence*, pages 301–320. Springer, 2003.

[125] E Zavitsanos, G Paliouras, GA Vouros, and S Petridis. Discovering Subsumption Hierarchies of Ontology Concepts from Text Corpora. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 402–408, 2007.

[126] V Spiliopoulos, G Vouros, and V Karkaletsis. On the Discovery of Subsumption Relations for the Alignment of Ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8:69–88, 2010.

[127] Ph Cimiano, A Hotho, and S Staab. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of Artificial Intelligence Research*, 24:305–339, 2005.

[128] P Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge, MA, 1988.

[129] CE Alchourrón, P Gärdenfors, and D Makinson. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

Glossary

Ontology Alignment – also called Ontology Matching, is the process of finding the mappings between entities from two different source ontologies. Depending on the purpose of a specific ontology alignment task, the entities can be *concepts*, *relations* or *instances*, and the relationships of the mappings can be *equivalence* as well as *is-a*, *part-of* or any other kind of relation.

Partial Reference Alignment (PRA) – is a set of correct mappings between entities from two ontologies.

Reference Alignment (RA) – is the complete set of correct mappings between entities from two ontologies.

Precision – measures how many of the mapping suggestions are correct in an ontology alignment result, which is defined as the number of correct suggestions divided by the number of suggestions.

Recall – measures how many of the correct mappings are found in a given ontology alignment result, which is defined as the number of correct suggestions divided by the number of correct mappings.

F-measure – is the weighted harmonic mean of precision and recall.

Resource Description Framework (RDF) – is a language that provides flexible mechanism for describing Web resources and the relationships among them.

Web Ontology Language (OWL) – is the standard Web Ontology Language proposed by W3C in 2004.

Avdelning, Institution
Division, Department

Institutionen för datavetenskap

Department of Computer and Information Science

Linköpings universitet

Datum
Date

2011-03-17

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ________________

Rapporttyp
Report category

☒ Licentiatavhandling
☐ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ ________________

ISBN
978-91-7393-233-2

ISRN
LiU-Tek-Lic–2011:7

Serietitel och serienummer
Title of series, numbering

ISSN
0280–7971

Linköping Studies in Science and Technology

Thesis No. 1468

URL för elektronisk version

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-64281

Titel
Title

Dealing with Missing Mappings and Structure in a Network of Ontologies

Författare
Author

Qiang Liu

Sammanfattning
Abstract

With the popularity of the World Wide Web, a large amount of data is generated and made available through the Internet everyday. To integrate and query this huge amount of heterogeneous data, the vision of Semantic Web has been recognized as a possible solution. One key technology for the Semantic Web is ontologies. Many ontologies have been developed in recent years. Meanwhile, due to the demand of applications using multiple ontologies, mappings between entities of these ontologies are generated as well, which leads to the generation of ontology networks consisting of ontologies and mappings between these ontologies. However, neither developing ontologies nor finding mappings between ontologies is an easy task. It may happen that the ontologies are not consistent or complete, or the mappings between these ontologies are not correct or complete, or the resulting ontology network is not consistent. This may lead to problems when they are used in semantically-enabled applications. In this thesis, we address two issues relevant to the quality of the mappings and the structure in the ontology network. The first issue deals with the missing mappings between networked ontologies. Assuming existing mappings between ontologies are correct, we investigate whether and how to use these existing mappings, to find more mappings between ontologies. We propose and test several strategies of using the given correct mappings to align ontologies. The second issue deals with the missing structure, in particular missing is-a relations, in networked ontologies. Based on the assumption that missing is-a relations are a kind of modeling defects, we propose an ontology debugging approach to tackle this issue. We develop an algorithm for detecting missing is-a relations in ontologies, as well as algorithms which assist the user in repairing by generating and recommending possible ways of repairing and executing the repairing. Based on this approach, we develop a system and test its use and performance.

Nyckelord
Keywords

Ontology, Ontology Engineering, Ontology Alignment, Ontology Debugging

Department of Computer and Information Science
Linköpings universitet

Linköpings Studies in Science and Technology
Faculty of Arts and Sciences – Licentiate Theses

No 17 **Vojin Plavsic:** Interleaved Processing of Non-Numerical Data Stored on a Cyclic Memory. (Available at: FOA, Box 1165, S-581 11 Linköping, Sweden. FOA Report B30062E)
No 28 **Arne Jönsson, Mikael Patel:** An Interactive Flowcharting Technique for Communicating and Realizing Algorithms, 1984.
No 29 **Johnny Eckerland:** Retargeting of an Incremental Code Generator, 1984.
No 48 **Henrik Nordin:** On the Use of Typical Cases for Knowledge-Based Consultation and Teaching, 1985.
No 52 **Zebo Peng:** Steps Towards the Formalization of Designing VLSI Systems, 1985.
No 60 **Johan Fagerström:** Simulation and Evaluation of Architecture based on Asynchronous Processes, 1985.
No 71 **Jalal Maleki:** ICONStraint, A Dependency Directed Constraint Maintenance System, 1987.
No 72 **Tony Larsson:** On the Specification and Verification of VLSI Systems, 1986.
No 73 **Ola Strömfors:** A Structure Editor for Documents and Programs, 1986.
No 74 **Christos Levcopoulos:** New Results about the Approximation Behavior of the Greedy Triangulation, 1986.
No 104 **Shamsul I. Chowdhury:** Statistical Expert Systems - a Special Application Area for Knowledge-Based Computer Methodology, 1987.
No 108 **Rober Bilos:** Incremental Scanning and Token-Based Editing, 1987.
No 111 **Hans Block:** SPORT-SORT Sorting Algorithms and Sport Tournaments, 1987.
No 113 **Ralph Rönnquist:** Network and Lattice Based Approaches to the Representation of Knowledge, 1987.
No 118 **Mariam Kamkar, Nahid Shahmehri:** Affect-Chaining in Program Flow Analysis Applied to Queries of Programs, 1987.
No 126 **Dan Strömberg:** Transfer and Distribution of Application Programs, 1987.
No 127 **Kristian Sandahl:** Case Studies in Knowledge Acquisition, Migration and User Acceptance of Expert Systems, 1987.
No 139 **Christer Bäckström:** Reasoning about Interdependent Actions, 1988.
No 140 **Mats Wirén:** On Control Strategies and Incrementality in Unification-Based Chart Parsing, 1988.
No 146 **Johan Hultman:** A Software System for Defining and Controlling Actions in a Mechanical System, 1988.
No 150 **Tim Hansen:** Diagnosing Faults using Knowledge about Malfunctioning Behavior, 1988.
No 165 **Jonas Löwgren:** Supporting Design and Management of Expert System User Interfaces, 1989.
No 166 **Ola Petersson:** On Adaptive Sorting in Sequential and Parallel Models, 1989.
No 174 **Yngve Larsson:** Dynamic Configuration in a Distributed Environment, 1989.
No 177 **Peter Åberg:** Design of a Multiple View Presentation and Interaction Manager, 1989.
No 181 **Henrik Eriksson:** A Study in Domain-Oriented Tool Support for Knowledge Acquisition, 1989.
No 184 **Ivan Rankin:** The Deep Generation of Text in Expert Critiquing Systems, 1989.
No 187 **Simin Nadjm-Tehrani:** Contributions to the Declarative Approach to Debugging Prolog Programs, 1989.
No 189 **Magnus Merkel:** Temporal Information in Natural Language, 1989.
No 196 **Ulf Nilsson:** A Systematic Approach to Abstract Interpretation of Logic Programs, 1989.
No 197 **Staffan Bonnier:** Horn Clause Logic with External Procedures: Towards a Theoretical Framework, 1989.
No 203 **Christer Hansson:** A Prototype System for Logical Reasoning about Time and Action, 1990.
No 212 **Björn Fjellborg:** An Approach to Extraction of Pipeline Structures for VLSI High-Level Synthesis, 1990.
No 230 **Patrick Doherty:** A Three-Valued Approach to Non-Monotonic Reasoning, 1990.
No 237 **Tomas Sokolnicki:** Coaching Partial Plans: An Approach to Knowledge-Based Tutoring, 1990.
No 250 **Lars Strömberg:** Postmortem Debugging of Distributed Systems, 1990.
No 253 **Torbjörn Näslund:** SLDFA-Resolution - Computing Answers for Negative Queries, 1990.
No 260 **Peter D. Holmes:** Using Connectivity Graphs to Support Map-Related Reasoning, 1991.
No 283 **Olof Johansson:** Improving Implementation of Graphical User Interfaces for Object-Oriented Knowledge- Bases, 1991.
No 298 **Rolf G Larsson:** Aktivitetsbaserad kalkylering i ett nytt ekonomisystem, 1991.
No 318 **Lena Srömbäck:** Studies in Extended Unification-Based Formalism for Linguistic Description: An Algorithm for Feature Structures with Disjunction and a Proposal for Flexible Systems, 1992.
No 319 **Mikael Pettersson:** DML-A Language and System for the Generation of Efficient Compilers from Denotational Specification, 1992.
No 326 **Andreas Kågedal:** Logic Programming with External Procedures: an Implementation, 1992.
No 328 **Patrick Lambrix:** Aspects of Version Management of Composite Objects, 1992.
No 333 **Xinli Gu:** Testability Analysis and Improvement in High-Level Synthesis Systems, 1992.
No 335 **Torbjörn Näslund:** On the Role of Evaluations in Iterative Development of Managerial Support Systems, 1992.
No 348 **Ulf Cederling:** Industrial Software Development - a Case Study, 1992.
No 352 **Magnus Morin:** Predictable Cyclic Computations in Autonomous Systems: A Computational Model and Implementation, 1992.
No 371 **Mehran Noghabai:** Evaluation of Strategic Investments in Information Technology, 1993.
No 378 **Mats Larsson:** A Transformational Approach to Formal Digital System Design, 1993.
No 380 **Johan Ringström:** Compiler Generation for Parallel Languages from Denotational Specifications, 1993.

No 381 **Michael Jansson:** Propagation of Change in an Intelligent Information System, 1993.
No 383 **Jonni Harrius:** An Architecture and a Knowledge Representation Model for Expert Critiquing Systems, 1993.
No 386 **Per Österling:** Symbolic Modelling of the Dynamic Environments of Autonomous Agents, 1993.
No 398 **Johan Boye:** Dependency-based Groudness Analysis of Functional Logic Programs, 1993.
No 402 **Lars Degerstedt:** Tabulated Resolution for Well Founded Semantics, 1993.
No 406 **Anna Moberg:** Satellitkontor - en studie av kommunikationsmönster vid arbete på distans, 1993.
No 414 **Peter Carlsson:** Separation av företagsledning och finansiering - fallstudier av företagsledarutköp ur ett agent-teoretiskt perspektiv, 1994.
No 417 **Camilla Sjöström:** Revision och lagreglering - ett historiskt perspektiv, 1994.
No 436 **Cecilia Sjöberg:** Voices in Design: Argumentation in Participatory Development, 1994.
No 437 **Lars Viklund:** Contributions to a High-level Programming Environment for a Scientific Computing, 1994.
No 440 **Peter Loborg:** Error Recovery Support in Manufacturing Control Systems, 1994.
FHS 3/94 **Owen Eriksson:** Informationssystem med verksamhetskvalitet - utvärdering baserat på ett verksamhetsinriktat och samskapande perspektiv, 1994.
FHS 4/94 **Karin Pettersson:** Informationssystemstrukturering, ansvarsfördelning och användarinflytande - En komparativ studie med utgångspunkt i två informationssystemstrategier, 1994.
No 441 **Lars Poignant:** Informationsteknologi och företagsetablering - Effekter på produktivitet och region, 1994.
No 446 **Gustav Fahl:** Object Views of Relational Data in Multidatabase Systems, 1994.
No 450 **Henrik Nilsson:** A Declarative Approach to Debugging for Lazy Functional Languages, 1994.
No 451 **Jonas Lind:** Creditor - Firm Relations: an Interdisciplinary Analysis, 1994.
No 452 **Martin Sköld:** Active Rules based on Object Relational Queries - Efficient Change Monitoring Techniques, 1994.
No 455 **Pär Carlshamre:** A Collaborative Approach to Usability Engineering: Technical Communicators and System Developers in Usability-Oriented Systems Development, 1994.
FHS 5/94 **Stefan Cronholm:** Varför CASE-verktyg i systemutveckling? - En motiv- och konsekvensstudie avseende arbetssätt och arbetsformer, 1994.
No 462 **Mikael Lindvall:** A Study of Traceability in Object-Oriented Systems Development, 1994.
No 463 **Fredrik Nilsson:** Strategi och ekonomisk styrning - En studie av Sandviks förvärv av Bahco Verktyg, 1994.
No 464 **Hans Olsén:** Collage Induction: Proving Properties of Logic Programs by Program Synthesis, 1994.
No 469 **Lars Karlsson:** Specification and Synthesis of Plans Using the Features and Fluents Framework, 1995.
No 473 **Ulf Söderman:** On Conceptual Modelling of Mode Switching Systems, 1995.
No 475 **Choong-ho Yi:** Reasoning about Concurrent Actions in the Trajectory Semantics, 1995.
No 476 **Bo Lagerström:** Successiv resultatavräkning av pågående arbeten. - Fallstudier i tre byggföretag, 1995.
No 478 **Peter Jonsson:** Complexity of State-Variable Planning under Structural Restrictions, 1995.
FHS 7/95 **Anders Avdic:** Arbetsintegrerad systemutveckling med kalkylprogram, 1995.
No 482 **Eva L Ragnemalm:** Towards Student Modelling through Collaborative Dialogue with a Learning Companion, 1995.
No 488 **Eva Toller:** Contributions to Parallel Multiparadigm Languages: Combining Object-Oriented and Rule-Based Programming, 1995.
No 489 **Erik Stoy:** A Petri Net Based Unified Representation for Hardware/Software Co-Design, 1995.
No 497 **Johan Herber:** Environment Support for Building Structured Mathematical Models, 1995.
No 498 **Stefan Svenberg:** Structure-Driven Derivation of Inter-Lingual Functor-Argument Trees for Multi-Lingual Generation, 1995.
No 503 **Hee-Cheol Kim:** Prediction and Postdiction under Uncertainty, 1995.
FHS 8/95 **Dan Fristedt:** Metoder i användning - mot förbättring av systemutveckling genom situationell metodkunskap och metodanalys, 1995.
FHS 9/95 **Malin Bergvall:** Systemförvaltning i praktiken - en kvalitativ studie avseende centrala begrepp, aktiviteter och ansvarsroller, 1995.
No 513 **Joachim Karlsson:** Towards a Strategy for Software Requirements Selection, 1995.
No 517 **Jakob Axelsson:** Schedulability-Driven Partitioning of Heterogeneous Real-Time Systems, 1995.
No 518 **Göran Forslund:** Toward Cooperative Advice-Giving Systems: The Expert Systems Experience, 1995.
No 522 **Jörgen Andersson:** Bilder av småföretagares ekonomistyrning, 1995.
No 538 **Staffan Flodin:** Efficient Management of Object-Oriented Queries with Late Binding, 1996.
No 545 **Vadim Engelson:** An Approach to Automatic Construction of Graphical User Interfaces for Applications in Scientific Computing, 1996.
No 546 **Magnus Werner :** Multidatabase Integration using Polymorphic Queries and Views, 1996.
FiF-a 1/96 **Mikael Lind:** Affärsprocessinriktad förändringsanalys - utveckling och tillämpning av synsätt och metod, 1996.
No 549 **Jonas Hallberg:** High-Level Synthesis under Local Timing Constraints, 1996.
No 550 **Kristina Larsen:** Förutsättningar och begränsningar för arbete på distans - erfarenheter från fyra svenska företag. 1996.
No 557 **Mikael Johansson:** Quality Functions for Requirements Engineering Methods, 1996.
No 558 **Patrik Nordling:** The Simulation of Rolling Bearing Dynamics on Parallel Computers, 1996.
No 561 **Anders Ekman:** Exploration of Polygonal Environments, 1996.
No 563 **Niclas Andersson:** Compilation of Mathematical Models to Parallel Code, 1996.
No 567 **Johan Jenvald:** Simulation and Data Collection in Battle Training, 1996.

No 575 **Niclas Ohlsson:** Software Quality Engineering by Early Identification of Fault-Prone Modules, 1996.
No 576 **Mikael Ericsson:** Commenting Systems as Design Support—A Wizard-of-Oz Study, 1996.
No 587 **Jörgen Lindström:** Chefers användning av kommunikationsteknik, 1996.
No 589 **Esa Falkenroth:** Data Management in Control Applications - A Proposal Based on Active Database Systems, 1996.
No 591 **Niclas Wahllöf:** A Default Extension to Description Logics and its Applications, 1996.
No 595 **Annika Larsson:** Ekonomisk Styrning och Organisatorisk Passion - ett interaktivt perspektiv, 1997.
No 597 **Ling Lin:** A Value-based Indexing Technique for Time Sequences, 1997.
No 598 **Rego Granlund:** C^3Fire - A Microworld Supporting Emergency Management Training, 1997.
No 599 **Peter Ingels:** A Robust Text Processing Technique Applied to Lexical Error Recovery, 1997.
No 607 **Per-Arne Persson:** Toward a Grounded Theory for Support of Command and Control in Military Coalitions, 1997.
No 609 **Jonas S Karlsson:** A Scalable Data Structure for a Parallel Data Server, 1997.
FiF-a 4 **Carita Åbom:** Videomötesteknik i olika affärssituationer - möjligheter och hinder, 1997.
FiF-a 6 **Tommy Wedlund**: Att skapa en företagsanpassad systemutvecklingsmodell - genom rekonstruktion, värdering och vidareutveckling i T50-bolag inom ABB, 1997.
No 615 **Silvia Coradeschi**: A Decision-Mechanism for Reactive and Coordinated Agents, 1997.
No 623 **Jan Ollinen:** Det flexibla kontorets utveckling på Digital - Ett stöd för multiflex? 1997.
No 626 **David Byers:** Towards Estimating Software Testability Using Static Analysis, 1997.
No 627 **Fredrik Eklund:** Declarative Error Diagnosis of GAPLog Programs, 1997.
No 629 **Gunilla Ivefors:** Krigsspel och Informationsteknik inför en oförutsägbar framtid, 1997**.**
No 631 **Jens-Olof Lindh:** Analysing Traffic Safety from a Case-Based Reasoning Perspective, 1997
No 639 **Jukka Mäki-Turja:**. Smalltalk - a suitable Real-Time Language, 1997.
No 640 **Juha Takkinen**: CAFE: Towards a Conceptual Model for Information Management in Electronic Mail, 1997.
No 643 **Man Lin**: Formal Analysis of Reactive Rule-based Programs, 1997.
No 653 **Mats Gustafsson**: Bringing Role-Based Access Control to Distributed Systems, 1997.
FiF-a 13 **Boris Karlsson:** Metodanalys för förståelse och utveckling av systemutvecklingsverksamhet. Analys och värdering av systemutvecklingsmodeller och dess användning, 1997.
No 674 **Marcus Bjäreland:** Two Aspects of Automating Logics of Action and Change - Regression and Tractability, 1998.
No 676 **Jan Håkegård**: Hierarchical Test Architecture and Board-Level Test Controller Synthesis, 1998.
No 668 **Per-Ove Zetterlund**: Normering av svensk redovisning - En studie av tillkomsten av Redovisningsrådets rekommendation om koncernredovisning (RR01:91), 1998.
No 675 **Jimmy Tjäder**: Projektledaren & planen - en studie av projektledning i tre installations- och systemutvecklingsprojekt, 1998.
FiF-a 14 **Ulf Melin**: Informationssystem vid ökad affärs- och processorientering - egenskaper, strategier och utveckling, 1998.
No 695 **Tim Heyer**: COMPASS: Introduction of Formal Methods in Code Development and Inspection, 1998.
No 700 **Patrik Hägglund:** Programming Languages for Computer Algebra, 1998.
FiF-a 16 **Marie-Therese Christiansson:** Inter-organisatorisk verksamhetsutveckling - metoder som stöd vid utveckling av partnerskap och informationssystem, 1998.
No 712 **Christina Wennestam:** Information om immateriella resurser. Investeringar i forskning och utveckling samt i personal inom skogsindustrin, 1998.
No 719 **Joakim Gustafsson:** Extending Temporal Action Logic for Ramification and Concurrency, 1998.
No 723 **Henrik André-Jönsson:** Indexing time-series data using text indexing methods, 1999.
No 725 **Erik Larsson:** High-Level Testability Analysis and Enhancement Techniques, 1998.
No 730 **Carl-Johan Westin:** Informationsförsörjning: en fråga om ansvar - aktiviteter och uppdrag i fem stora svenska organisationers operativa informationsförsörjning, 1998.
No 731 **Åse Jansson:** Miljöhänsyn - en del i företags styrning, 1998.
No 733 **Thomas Padron-McCarthy:** Performance-Polymorphic Declarative Queries, 1998.
No 734 **Anders Bäckström:** Värdeskapande kreditgivning - Kreditriskhantering ur ett agentteoretiskt perspektiv, 1998.
FiF-a 21 **Ulf Seigerroth:** Integration av förändringsmetoder - en modell för välgrundad metodintegration, 1999.
FiF-a 22 **Fredrik Öberg:** Object-Oriented Frameworks - A New Strategy for Case Tool Development, 1998.
No 737 **Jonas Mellin:** Predictable Event Monitoring, 1998.
No 738 **Joakim Eriksson:** Specifying and Managing Rules in an Active Real-Time Database System, 1998.
FiF-a 25 **Bengt E W Andersson:** Samverkande informationssystem mellan aktörer i offentliga åtaganden - En teori om aktörsarenor i samverkan om utbyte av information, 1998.
No 742 **Pawel Pietrzak:** Static Incorrectness Diagnosis of CLP (FD), 1999.
No 748 **Tobias Ritzau:** Real-Time Reference Counting in RT-Java, 1999.
No 751 **Anders Ferntoft:** Elektronisk affärskommunikation - kontaktkostnader och kontaktprocesser mellan kunder och leverantörer på producentmarknader, 1999.
No 752 **Jo Skåmedal:** Arbete på distans och arbetsformens påverkan på resor och resmönster, 1999.
No 753 **Johan Alvehus:** Mötets metaforer. En studie av berättelser om möten, 1999.
No 754 **Magnus Lindahl:** Bankens villkor i låneavtal vid kreditgivning till högt belånade företagsförvärv: En studie ur ett agentteoretiskt perspektiv, 2000.

No 766 **Martin V. Howard:** Designing dynamic visualizations of temporal data, 1999.
No 769 **Jesper Andersson:** Towards Reactive Software Architectures, 1999.
No 775 **Anders Henriksson:** Unique kernel diagnosis, 1999.
FiF-a 30 **Pär J. Ågerfalk:** Pragmatization of Information Systems - A Theoretical and Methodological Outline, 1999.
No 787 **Charlotte Björkegren:** Learning for the next project - Bearers and barriers in knowledge transfer within an organisation, 1999.
No 788 **Håkan Nilsson:** Informationsteknik som drivkraft i granskningsprocessen - En studie av fyra revisionsbyråer, 2000.
No 790 **Erik Berglund:** Use-Oriented Documentation in Software Development, 1999.
No 791 **Klas Gäre:** Verksamhetsförändringar i samband med IS-införande, 1999.
No 800 **Anders Subotic:** Software Quality Inspection, 1999.
No 807 **Svein Bergum**: Managerial communication in telework, 2000.
No 809 **Flavius Gruian:** Energy-Aware Design of Digital Systems, 2000.
FiF-a 32 **Karin Hedström:** Kunskapsanvändning och kunskapsutveckling hos verksamhetskonsulter - Erfarenheter från ett FOU-samarbete, 2000.
No 808 **Linda Askenäs:** Affärssystemet - En studie om teknikens aktiva och passiva roll i en organisation, 2000.
No 820 **Jean Paul Meynard:** Control of industrial robots through high-level task programming, 2000.
No 823 **Lars Hult:** Publika Gränsytor - ett designexempel, 2000.
No 832 **Paul Pop:** Scheduling and Communication Synthesis for Distributed Real-Time Systems, 2000.
FiF-a 34 **Göran Hultgren:** Nätverksinriktad Förändringsanalys - perspektiv och metoder som stöd för förståelse och utveckling av affärsrelationer och informationssystem, 2000.
No 842 **Magnus Kald:** The role of management control systems in strategic business units, 2000.
No 844 **Mikael Cäker:** Vad kostar kunden? Modeller för intern redovisning, 2000.
FiF-a 37 **Ewa Braf**: Organisationers kunskapsverksamheter - en kritisk studie av ”knowledge management”, 2000.
FiF-a 40 **Henrik Lindberg:** Webbaserade affärsprocesser - Möjligheter och begränsningar, 2000.
FiF-a 41 **Benneth Christiansson:** Att komponentbasera informationssystem - Vad säger teori och praktik?, 2000.
No. 854 **Ola Pettersson:** Deliberation in a Mobile Robot, 2000.
No 863 **Dan Lawesson**: Towards Behavioral Model Fault Isolation for Object Oriented Control Systems, 2000.
No 881 **Johan Moe:** Execution Tracing of Large Distributed Systems, 2001.
No 882 **Yuxiao Zhao:** XML-based Frameworks for Internet Commerce and an Implementation of B2B e-procurement, 2001.
No 890 **Annika Flycht-Eriksson:** Domain Knowledge Management in Information-providing Dialogue systems, 2001.
FiF-a 47 **Per-Arne Segerkvist**: Webbaserade imaginära organisationers samverkansformer: Informationssystemarkitektur och aktörssamverkan som förutsättningar för affärsprocesser, 2001.
No 894 **Stefan Svarén:** Styrning av investeringar i divisionaliserade företag - Ett koncernperspektiv, 2001.
No 906 **Lin Han**: Secure and Scalable E-Service Software Delivery, 2001.
No 917 **Emma Hansson:** Optionsprogram för anställda - en studie av svenska börsföretag, 2001.
No 916 **Susanne Odar:** IT som stöd för strategiska beslut, en studie av datorimplementerade modeller av verksamhet som stöd för beslut om anskaffning av JAS 1982, 2002.
FiF-a-49 **Stefan Holgersson:** IT-system och filtrering av verksamhetskunskap - kvalitetsproblem vid analyser och beslutsfattande som bygger på uppgifter hämtade från polisens IT-system, 2001.
FiF-a-51 **Per Oscarsson:** Informationssäkerhet i verksamheter - begrepp och modeller som stöd för förståelse av informationssäkerhet och dess hantering, 2001.
No 919 **Luis Alejandro Cortes:** A Petri Net Based Modeling and Verification Technique for Real-Time Embedded Systems, 2001.
No 915 **Niklas Sandell:** Redovisning i skuggan av en bankkris - Värdering av fastigheter. 2001.
No 931 **Fredrik Elg:** Ett dynamiskt perspektiv på individuella skillnader av heuristisk kompetens, intelligens, mentala modeller, mål och konfidens i kontroll av mikrovärlden Moro, 2002.
No 933 **Peter Aronsson:** Automatic Parallelization of Simulation Code from Equation Based Simulation Languages, 2002.
No 938 **Bourhane Kadmiry**: Fuzzy Control of Unmanned Helicopter, 2002.
No 942 **Patrik Haslum**: Prediction as a Knowledge Representation Problem: A Case Study in Model Design, 2002.
No 956 **Robert Sevenius:** On the instruments of governance - A law & economics study of capital instruments in limited liability companies, 2002.
FiF-a 58 **Johan Petersson:** Lokala elektroniska marknadsplatser - informationssystem för platsbundna affärer, 2002.
No 964 **Peter Bunus:** Debugging and Structural Analysis of Declarative Equation-Based Languages, 2002.
No 973 **Gert Jervan:** High-Level Test Generation and Built-In Self-Test Techniques for Digital Systems, 2002.
No 958 **Fredrika Berglund:** Management Control and Strategy - a Case Study of Pharmaceutical Drug Development, 2002.
FiF-a 61 **Fredrik Karlsson:** Meta-Method for Method Configuration - A Rational Unified Process Case, 2002.
No 985 **Sorin Manolache:** Schedulability Analysis of Real-Time Systems with Stochastic Task Execution Times, 2002.
No 982 **Diana Szentiványi:** Performance and Availability Trade-offs in Fault-Tolerant Middleware, 2002.
No 989 **Iakov Nakhimovski:** Modeling and Simulation of Contacting Flexible Bodies in Multibody Systems, 2002.
No 990 **Levon Saldamli:** PDEModelica - Towards a High-Level Language for Modeling with Partial Differential Equations, 2002.
No 991 **Almut Herzog:** Secure Execution Environment for Java Electronic Services, 2002.
No 999 **Jon Edvardsson:** Contributions to Program- and Specification-based Test Data Generation, 2002.
No 1000 **Anders Arpteg:** Adaptive Semi-structured Information Extraction, 2002.

No 1001 **Andrzej Bednarski:** A Dynamic Programming Approach to Optimal Retargetable Code Generation for Irregular Architectures, 2002.
No 988 **Mattias Arvola:** Good to use! : Use quality of multi-user applications in the home, 2003.
FiF-a 62 **Lennart Ljung:** Utveckling av en projektivitetsmodell - om organisationers förmåga att tillämpa projektarbetsformen, 2003.
No 1003 **Pernilla Qvarfordt:** User experience of spoken feedback in multimodal interaction, 2003.
No 1005 **Alexander Siemers:** Visualization of Dynamic Multibody Simulation With Special Reference to Contacts, 2003.
No 1008 **Jens Gustavsson:** Towards Unanticipated Runtime Software Evolution, 2003.
No 1010 **Calin Curescu:** Adaptive QoS-aware Resource Allocation for Wireless Networks, 2003.
No 1015 **Anna Andersson:** Management Information Systems in Process-oriented Healthcare Organisations, 2003.
No 1018 **Björn Johansson:** Feedforward Control in Dynamic Situations, 2003.
No 1022 **Traian Pop:** Scheduling and Optimisation of Heterogeneous Time/Event-Triggered Distributed Embedded Systems, 2003.
FiF-a 65 **Britt-Marie Johansson:** Kundkommunikation på distans - en studie om kommunikationsmediets betydelse i affärstransaktioner, 2003.
No 1024 **Aleksandra Tešanovic:** Towards Aspectual Component-Based Real-Time System Development, 2003.
No 1034 **Arja Vainio-Larsson**: Designing for Use in a Future Context - Five Case Studies in Retrospect, 2003.
No 1033 **Peter Nilsson:** Svenska bankers redovisningsval vid reservering för befarade kreditförluster - En studie vid införandet av nya redovisningsregler, 2003.
FiF-a 69 **Fredrik Ericsson:** Information Technology for Learning and Acquiring of Work Knowledge, 2003.
No 1049 **Marcus Comstedt:** Towards Fine-Grained Binary Composition through Link Time Weaving, 2003.
No 1052 **Åsa Hedenskog:** Increasing the Automation of Radio Network Control, 2003.
No 1054 **Claudiu Duma:** Security and Efficiency Tradeoffs in Multicast Group Key Management, 2003.
FiF-a 71 **Emma Eliason:** Effektanalys av IT-systems handlingsutrymme, 2003.
No 1055 **Carl Cederberg:** Experiments in Indirect Fault Injection with Open Source and Industrial Software, 2003.
No 1058 **Daniel Karlsson:** Towards Formal Verification in a Component-based Reuse Methodology, 2003.
FiF-a 73 **Anders Hjalmarsson:** Att etablera och vidmakthålla förbättringsverksamhet - behovet av koordination och interaktion vid förändring av systemutvecklingsverksamheter, 2004.
No 1079 **Pontus Johansson:** Design and Development of Recommender Dialogue Systems, 2004.
No 1084 **Charlotte Stoltz:** Calling for Call Centres - A Study of Call Centre Locations in a Swedish Rural Region, 2004.
FiF-a 74 **Björn Johansson:** Deciding on Using Application Service Provision in SMEs, 2004.
No 1094 **Genevieve Gorrell:** Language Modelling and Error Handling in Spoken Dialogue Systems, 2004.
No 1095 **Ulf Johansson:** Rule Extraction - the Key to Accurate and Comprehensible Data Mining Models, 2004.
No 1099 **Sonia Sangari:** Computational Models of Some Communicative Head Movements, 2004.
No 1110 **Hans Nässla:** Intra-Family Information Flow and Prospects for Communication Systems, 2004.
No 1116 **Henrik Sällberg:** On the value of customer loyalty programs - A study of point programs and switching costs, 2004.
FiF-a 77 **Ulf Larsson:** Designarbete i dialog - karaktärisering av interaktionen mellan användare och utvecklare i en systemutvecklingsprocess, 2004.
No 1126 **Andreas Borg:** Contribution to Management and Validation of Non-Functional Requirements, 2004.
No 1127 **Per-Ola Kristensson:** Large Vocabulary Shorthand Writing on Stylus Keyboard, 2004.
No 1132 **Pär-Anders Albinsson:** Interacting with Command and Control Systems: Tools for Operators and Designers, 2004.
No 1130 **Ioan Chisalita:** Safety-Oriented Communication in Mobile Networks for Vehicles, 2004.
No 1138 **Thomas Gustafsson:** Maintaining Data Consistency in Embedded Databases for Vehicular Systems, 2004.
No 1149 **Vaida Jakoniené:** A Study in Integrating Multiple Biological Data Sources, 2005.
No 1156 **Abdil Rashid Mohamed**: High-Level Techniques for Built-In Self-Test Resources Optimization, 2005.
No 1162 **Adrian Pop:** Contributions to Meta-Modeling Tools and Methods, 2005.
No 1165 **Fidel Vascós Palacios:** On the information exchange between physicians and social insurance officers in the sick leave process: an Activity Theoretical perspective, 2005.
FiF-a 84 **Jenny Lagsten:** Verksamhetsutvecklande utvärdering i informationssystemprojekt, 2005.
No 1166 **Emma Larsdotter Nilsson:** Modeling, Simulation, and Visualization of Metabolic Pathways Using Modelica, 2005.
No 1167 **Christina Keller:** Virtual Learning Environments in higher education. A study of students' acceptance of educational technology, 2005.
No 1168 **Cécile Åberg:** Integration of organizational workflows and the Semantic Web, 2005.
FiF-a 85 **Anders Forsman:** Standardisering som grund för informationssamverkan och IT-tjänster - En fallstudie baserad på trafikinformationstjänsten RDS-TMC, 2005.
No 1171 **Yu-Hsing Huang:** A systemic traffic accident model, 2005.
FiF-a 86 **Jan Olausson:** Att modellera uppdrag - grunder för förståelse av processinriktade informationssystem i transaktionsintensiva verksamheter, 2005.
No 1172 **Petter Ahlström**: Affärsstrategier för seniorbostadsmarknaden, 2005.
No 1183 **Mathias Cöster**: Beyond IT and Productivity - How Digitization Transformed the Graphic Industry, 2005.
No 1184 **Åsa Horzella**: Beyond IT and Productivity - Effects of Digitized Information Flows in Grocery Distribution, 2005.
No 1185 **Maria Kollberg**: Beyond IT and Productivity - Effects of Digitized Information Flows in the Logging Industry, 2005.
No 1190 **David Dinka**: Role and Identity - Experience of technology in professional settings, 2005.
No 1191 **Andreas Hansson**: Increasing the Storage Capacity of Recursive Auto-associative Memory by Segmenting Data, 2005.

No 1192 **Nicklas Bergfeldt:** Towards Detached Communication for Robot Cooperation, 2005.
No 1194 **Dennis Maciuszek:** Towards Dependable Virtual Companions for Later Life, 2005.
No 1204 **Beatrice Alenljung:** Decision-making in the Requirements Engineering Process: A Human-centered Approach, 2005.
No 1206 **Anders Larsson:** System-on-Chip Test Scheduling and Test Infrastructure Design, 2005.
No 1207 **John Wilander:** Policy and Implementation Assurance for Software Security, 2005.
No 1209 **Andreas Käll:** Översättningar av en managementmodell - En studie av införandet av Balanced Scorecard i ett landsting, 2005.
No 1225 **He Tan:** Aligning and Merging Biomedical Ontologies, 2006.
No 1228 **Artur Wilk:** Descriptive Types for XML Query Language Xcerpt, 2006.
No 1229 **Per Olof Pettersson:** Sampling-based Path Planning for an Autonomous Helicopter, 2006.
No 1231 **Kalle Burbeck:** Adaptive Real-time Anomaly Detection for Safeguarding Critical Networks, 2006.
No 1233 **Daniela Mihailescu:** Implementation Methodology in Action: A Study of an Enterprise Systems Implementation Methodology, 2006.
No 1244 **Jörgen Skågeby:** Public and Non-public gifting on the Internet, 2006.
No 1248 **Karolina Eliasson:** The Use of Case-Based Reasoning in a Human-Robot Dialog System, 2006.
No 1263 **Misook Park-Westman:** Managing Competence Development Programs in a Cross-Cultural Organisation - What are the Barriers and Enablers, 2006.
FiF-a 90 **Amra Halilovic:** Ett praktikperspektiv på hantering av mjukvarukomponenter, 2006.
No 1272 **Raquel Flodström**: A Framework for the Strategic Management of Information Technology, 2006.
No 1277 **Viacheslav Izosimov:** Scheduling and Optimization of Fault-Tolerant Embedded Systems, 2006.
No 1283 **Håkan Hasewinkel:** A Blueprint for Using Commercial Games off the Shelf in Defence Training, Education and Research Simulations, 2006.
FiF-a 91 **Hanna Broberg:** Verksamhetsanpassade IT-stöd - Designteori och metod, 2006.
No 1286 **Robert Kaminski:** Towards an XML Document Restructuring Framework, 2006.
No 1293 **Jiri Trnka:** Prerequisites for data sharing in emergency management, 2007.
No 1302 **Björn Hägglund:** A Framework for Designing Constraint Stores, 2007.
No 1303 **Daniel Andreasson**: Slack-Time Aware Dynamic Routing Schemes for On-Chip Networks, 2007.
No 1305 **Magnus Ingmarsson:** Modelling User Tasks and Intentions for Service Discovery in Ubiquitous Computing, 2007.
No 1306 **Gustaf Svedjemo**: Ontology as Conceptual Schema when Modelling Historical Maps for Database Storage, 2007.
No 1307 **Gianpaolo Conte**: Navigation Functionalities for an Autonomous UAV Helicopter, 2007.
No 1309 **Ola Leifler**: User-Centric Critiquing in Command and Control: The DKExpert and ComPlan Approaches, 2007.
No 1312 **Henrik Svensson**: Embodied simulation as off-line representation, 2007.
No 1313 **Zhiyuan He:** System-on-Chip Test Scheduling with Defect-Probability and Temperature Considerations, 2007.
No 1317 **Jonas Elmqvist:** Components, Safety Interfaces and Compositional Analysis, 2007.
No 1320 **Håkan Sundblad:** Question Classification in Question Answering Systems, 2007.
No 1323 **Magnus Lundqvist**: Information Demand and Use: Improving Information Flow within Small-scale Business Contexts, 2007.
No 1329 **Martin Magnusson:** Deductive Planning and Composite Actions in Temporal Action Logic, 2007.
No 1331 **Mikael Asplund:** Restoring Consistency after Network Partitions, 2007.
No 1332 **Martin Fransson:** Towards Individualized Drug Dosage - General Methods and Case Studies, 2007.
No 1333 **Karin Camara:** A Visual Query Language Served by a Multi-sensor Environment, 2007.
No 1337 **David Broman:** Safety, Security, and Semantic Aspects of Equation-Based Object-Oriented Languages and Environments, 2007.
No 1339 **Mikhail Chalabine:** Invasive Interactive Parallelization, 2007.
No 1351 **Susanna Nilsson:** A Holistic Approach to Usability Evaluations of Mixed Reality Systems, 2008.
No 1353 **Shanai Ardi:** A Model and Implementation of a Security Plug-in for the Software Life Cycle, 2008.
No 1356 **Erik Kuiper:** Mobility and Routing in a Delay-tolerant Network of Unmanned Aerial Vehicles, 2008.
No 1359 **Jana Rambusch**: Situated Play, 2008.
No 1361 **Martin Karresand:** Completing the Picture - Fragments and Back Again, 2008.
No 1363 **Per Nyblom:** Dynamic Abstraction for Interleaved Task Planning and Execution, 2008.
No 1371 **Fredrik Lantz:**Terrain Object Recognition and Context Fusion for Decision Support, 2008.
No 1373 **Martin Östlund:** Assistance Plus: 3D-mediated Advice-giving on Pharmaceutical Products, 2008.
No 1381 **Håkan Lundvall:** Automatic Parallelization using Pipelining for Equation-Based Simulation Languages, 2008.
No 1386 **Mirko Thorstensson:** Using Observers for Model Based Data Collection in Distributed Tactical Operations, 2008.
No 1387 **Bahlol Rahimi:** Implementation of Health Information Systems, 2008.
No 1392 **Maria Holmqvist:** Word Alignment by Re-using Parallel Phrases, 2008.
No 1393 **Mattias Eriksson**: Integrated Software Pipelining, 2009.
No 1401 **Annika Öhgren:** Towards an Ontology Development Methodology for Small and Medium-sized Enterprises, 2009.
No 1410 **Rickard Holsmark:** Deadlock Free Routing in Mesh Networks on Chip with Regions, 2009.
No 1421 **Sara Stymne:** Compound Processing for Phrase-Based Statistical Machine Translation, 2009.
No 1427 **Tommy Ellqvist**: Supporting Scientific Collaboration through Workflows and Provenance, 2009.
No 1450 **Fabian Segelström:** Visualisations in Service Design, 2010.
No 1459 **Min Bao**: System Level Techniques for Temperature-Aware Energy Optimization, 2010.
No 1468 **Qiang Liu**: Dealing with Missing Mappings and Structure in a Network of Ontologies, 2011.