Contexts and Ontologies: Theory, Practice and Applications

C&O-2006

Papers from the ECAI Workshop

Introduction

During the last decade, there has been a line of successful series of workshops and conferences on the development and application of contexts and ontologies. Early workshops were focused mostly on identifying what contexts and ontologies are, and how they can be formalized and exploited. In more recent years with the emergence of distributed systems (e.g., P2P systems, Semantic Web) the focus of workshops shifted towards issues of practical applications, such as semantic integration and coordination among information sources, where both contexts and ontologies were applied as promising solutions. However, few, if any, of these meetings have focused on combining the themes of ontologies and contexts and discussed them as complementary disciplines.

The goal of this workshop is to bring people from the context and ontology communities together to discuss the approaches they use for information integration. The workshop promotes the cross-fertilization and exchange of ideas (e.g., what are the commonalities and differences in the methods, which of the methods from one community can be successfully adopted by the other community). C&O-2005 revealed multiple perspectives on combinations of contexts and ontologies. One perspective views an ontology as an explicit encoding of a domain model that may be shared and reused, while a context may be viewed as an explicit encoding of a domain model that is expected to be local and may contain one partys subjective view of the domain. This workshop further explores this perspective as well as other perspectives and aims to make more progress in leveraging increased communication between the context and ontology communities. The workshop is open to technical areas of interest between contexts and ontologies, with anticipated focus on:

- approaches to semantic heterogeneity that utilize multiple contexts and ontologies;
- analysis and understanding of technical problems related to combination of contexts and ontologies from theoretical, practical and application perspectives.

We received 28 submissions for the workshop. The proceedings of the workshop contain long and short papers. Out of the submissions, 9 were selected as long papers for oral presentation and 11 were selected as short papers for poster presentation. Further information about the C&O workshops can be found at: `http://www.c-and-o.net/`

Acknowledgments. We thank all members of the program committee, additional reviewers, authors, invited speakers and local organizers for their efforts. We appreciate support from the Knowledge Web FP6 Network of Excellence.

Pavel Shvaiko
Jérôme Euzenat
Alain Léger
Deborah L. McGuinness
Holger Wache

August 2006

Organization

Program Chairs

Pavel Shvaiko, University of Trento, Italy (Chair)
Jérôme Euzenat, INRIA Rhône-Alpes, France (Cochair)
Alain Léger, France Telecom R&D, France (Cochair)
Deborah L. McGuinness, Stanford University, USA (Cochair)
Holger Wache, Vrije Universiteit Amsterdam, Netherlands (Cochair)

Program Committee

Raffaele de Amicis, GraphiTech, Italy
Alex Borgida, Rutgers University, USA
Patrick Brezillon, University Paris, France
Doina Caragea, Iowa State University, USA
Anind Dey, Carnegie Mellon University, USA
Avigdor Gal, Technion, Israel
Jérôme Euzenat, INRIA Rhône-Alpes, France
Richard Fikes, Stanford University, USA
Aykut Firat, Notheastern University, USA
Fausto Giunchiglia, University of Trento, Italy
Frank van Harmelen, Vrije Universiteit Amsterdam, Netherlands
Michael Huhns, University of South Carolina, USA
Yannis Kalfoglou, University of Southampton, UK
Vipul Kashyap, Clinical Informatics R&D, Partners HealthCare System, USA
Ruediger Klein, DaimlerChrysler, Germany
David Leake, Indiana University, USA
Alain Léger, France Telecom R&D, France
Selene Makarios, Stanford University, USA
Maurizio Marchese, University of Trento, Italy
Deborah L. McGuinness, Stanford University, USA
Natalya Noy, Stanford University, USA
Leo Obrst, MITRE, USA
Dimitris Plexousakis, University of Crete, Greece
Michel Plu, France Telecom R&D, France
Fano Ramparany, France Telecom R&D, France
Chantal Reynaud, Universite Paris-Sud, France
David Robertson, University of Edinburgh, UK
Aviv Segev, Technion, Israel
Pavel Shvaiko, University of Trento, Italy
Heiner Stuckenschmidt, University of Mannheim, Germany
York Sure, University of Karlsruhe, Germany
Holger Wache, Vrije Universiteit Amsterdam, Netherlands
Christopher Welty, IBM Research, USA

Additional Reviewers

Maxym Mykhalchuk, University of Trento, Italy
Mikalai Yatskevich, University of Trento, Italy

Table of Contents

PART 1 - Technical Papers

PART 2 - Posters and Statements of Interest

Corpus-Driven Contextualized Categorization

Tony Veale and **Yanfen Hao**[1]

Abstract. Ontologies strive to offer a interconnected, hierarchical systems of categories to guide our actions in a complex world. But the boundaries of these categories are highly context-dependent, and what constitutes a prototypical category member in one context may be atypical or unrepresentative in another. In this paper we outline a dynamic, trainable, bottom-up view of category structure based on context-sensitive corpus analysis. By learning from corpora about how people creatively actually use categories in different contexts, we can train our ontologies to creatively adapt themselves to these contexts.

1 INTRODUCTION

An ontology is a system of inter-connected categories that collectively provide a structured representation of a given domain. As such, an ontology serves as the conceptual bedrock against which domain meanings are constructed, manipulated and interpreted. However, this fundamental role of the ontology should not blind us to the fact that much of what an ontology attempts to model, via its category structure, is not static but dynamic, making the use of these categories highly sensitive to context. Consider that many categories in a language-oriented ontology, like Genius, Fool, Hero, Villain, Expert, Hunter, and so on, possess subjective membership criteria that change from user to user, and from context to context. Are politicians fools, villains or schemers? Are firemen heroes or workmen? Are scientists experts or geniuses?

Since top-down definitions of membership criteria will always seem brittle or inadequate in some contexts, it seems best to allow contexts to define their own criteria, bottom-up. In other words, we need to establish a contextual ontology [10] based category structure, which not only preserves the common view of concepts, but also keeps the local perspective of domains. For language-oriented ontologies, like WordNet [6] (a flawed, lightweight ontology to be sure, but an ontology none the less), HowNet [1] and, to some extent, Cyc [5], the context of usage can conveniently be captured via a large corpus of representative texts. A corpus-based approach to determining category membership allows us to structure the middle and lower layers of an ontology according to how words and concepts are actually used in a particular domain. In short, a corpus-based approach supports an extremely flexible, non-classical view of category structure, one that views category membership as a graded rather than binary notion [4], and one in which concepts can fluidly move (via metaphor) from one category to another [2]. In this current work, we use the ability to support metaphoric reasoning as the yardstick against which ontological flexibility should be measured.

Of course, this fluidity does not sit well with conventional perspectives on ontological structure, as represented by the ontologies of [1,5,6]. In this paper we look at one conventional ontology, the HowNet system of [1], which is a large-scale bilingual lexical ontology for words and their meanings in both Chinese and English. In many respects, HowNet is similar to the WordNet lexical ontology for English [6], though in contrast to WordNet, HowNet provides an explicit, if sparse, propositional semantics for each of the word-concepts it defines. Complementing this frame-like semantics, in which concepts are defined in terms of actions, case-roles and fillers, is a taxonomic backbone that seems rather impoverished when compared to that of WordNet. HowNet is essentially an ontology of "Being" rather than an ontology of "Doing" which is to say that it defines concepts according to conventional kinds like *human*, *animal*, *tool* and so on - rather than according to how specific concepts actually behave in context. However, we describe in section 2 how HowNet's propositional semantics can be used to automatically derive an ontology of "Doing" to replace HowNet's rather shallow taxonomy of conventional categories [8]. Once in place, we demonstrate how this new system of derived categories can be made contextually sensitive by defining their membership criteria in statistical, corpus-based terms, to create a fluid system of membership akin to the Slipnets of Hofstadter [3]. Once sensitized in this way, the ontology can be moved with ease from one context to another simply by replacing the underlying corpus.

2 ONTOLOGIES OF "BEING" AND "DOING"

HowNet and WordNet each reflect a different view of semantic organization. WordNet [7] is *differential* in nature: rather than attempting to express the meaning of a word explicitly, WordNet instead differentiates words with different meanings by placing them in different synonym sets, or *synsets*, and further differentiates these synsets from one another by assigning them to different positions of a taxonomy. In contrast, HowNet is *constructive* in nature. It does not provide a human-oriented textual gloss for each lexical concept, but instead composes sememes from a less discriminating taxonomy to provide a semantic representation for each word sense. For example, HowNet defines the lexical concept *surgeon*|医生 as follows:

(1)*surgeon*|医生 {*human*|人 :*HostOf*={*Occupation*|职位}
domain={*medical*|医}}, {*doctor*|医治:*agent*={~}}}

which can be glossed thus: "a surgeon is a human, with an occupation in the medical domain, who acts as an agent of a doctoring activity" (the {~} here serves to indicate the placement of the concept within its associated propositional structure). We see a similar structure employed by HowNet for the lexical concept *repairman*|修理工:

[1] School of Computer Science and Informatics, University College Dublin, Ireland, email: {tony.veale, yanfen.hao}@ucd.ie

(2)*repairman*|修理工 {*human*|人:*HostOf*={*Occupation*|职位}, {*repair*|修理:*agent*={∼}}}

Note that the impoverished nature of HowNet's taxonomy means that over 3000 different concepts are forced to share the immediate hypernym *human*|人. However, *human*|人 merely states, very generally, what a repairman is, rather than what a repairman does. Fortunately, HowNet also organizes its verb entries taxonomically, and so we find the verbs *doctor*|医治 and *repair*|修理 organized under the hypernym *resume*|恢复 (the logic being, one supposes, that "doctoring" and "repairing" both involve a resumption of an earlier, better state). This similarity of verbs, combined with an identicality of case-roles (both surgeon and repairman are agents of their respective activities), allows us to abstract out a new taxonomy, based on the behaviour rather than the general type of these entities.

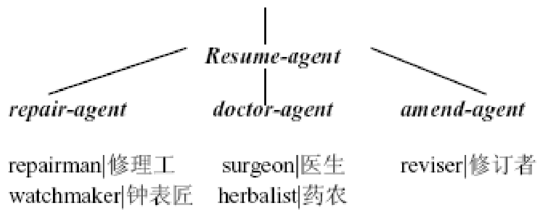

Figure 1. A new 3-level abstraction hierarchy derived from verb/role combinations.

Figure 1 illustrates the creation of such a taxonomy, whose categories represent a yoking of verbs to specific case-roles, such as *repair-agent* and *amend-agent*, and whose category members are those HowNet concepts defined using these verbs and roles. The category-hopping nature of metaphor is now rather easily construed as a combination of generalization and re-specialization operations, in which one moves from one category to another by first passing through a common super-category like *resume-agent*. Thus, a surgeon can be seen as a repairman or a watchmaker, while a reviser of texts (an editor) can sometimes be seen as a surgeon. These metaphors make sense not because each is a human, but because each restores a better state.

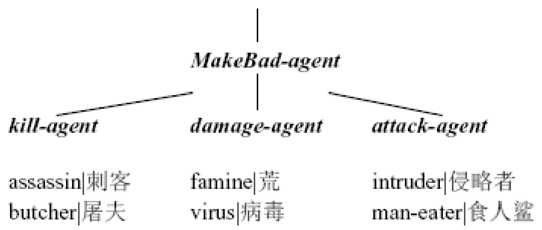

Figure 2. Newly derived HowNet categories may contain a diverse range of concepts.

Of course, this Aristotelian view of metaphor as an abstract "carrying-over" (the etymological origin of the word "metaphor") can only be valid if concepts are ontologized by what they do, rather than by what they are (as is typically the case, in both WordNet and HowNet, and even Cyc [6]). Otherwise, metaphor could never operate between semantically distant concepts, which it plainly does. For instance, figure 2 illustrates the derived taxonomy for HowNet concepts that are defined as agents of the verbs "kill", "damage" and "attack", each a specialization of the abstract verb *MakeBad* in HowNet. We see in this taxonomy the potential for famines to be metaphorically viewed as butchers and assassins, and for viruses to be seen as deadly intruders, or even man-eaters.

3 DERIVING FLUID CATEGORY STRUCTURES

An ontology of "doing" begs a number of obvious questions about the nature of categorization. For instance, is every concept that kills an equally representative member of the category *kill-agent*? Is movement always allowed between any two categories that share a common abstraction like *MakeBad-agent*, or is movement limited to certain members only, and in certain directions? When a concept moves from its conventional category to another, how is its degree of membership in this new category to be assessed? In this section we address this key issue of obtaining fluid category structure.

There are two major approaches in the community of automatic acquisition of taxonomies. One approach is based on the distributional hypothesis made by Harris[11], in which he believes that word terms are similar if they have similar linguistic contexts. For instance, Hindle[12] clusters nouns according to their contextual attributes, such as the co-occurrence of nouns with verbs as subjects or objects. Steffen Staab[13] also extracts context information (verb/subject dependencies, verb/object dependencies, e.g.) about a certain term from corpus and applies a Formal Concept Analysis to generate a lattice that is finally transformed into a partial order closer to a concept hierarchy. Another major approach is on the basis of investigating the ontological relations such as is-a relation, part-of relation, e.g. via the corpus. Hearst[14] is a representative of this field. However, it seems that these approaches still result in binary and static taxonomies because they all apply the threshold to the category or the concept architecture to determine whether or not a word concept belongs to it. In our approach, we also follow Harris[11]'s distributional hypothesis to investigate the contextual attributes, particularly, the behavior of nouns. The difference is that we apply Lakoff[4]'s category theory to assign the graded membership to nouns within a category rather than simply grouping them into classes according to their contextual attributes or ontological relations.

Following Lakoff [4], every category will possess a prototype, a member that is highly representative of the category as a whole. Such prototypes are often lexicalized in simple terms; for instance, "killer" will be a highly representative of *kill-agent*, while the Chinese translation "杀手" is a composition of "killing" (杀) and "expert" (手). However, many categories like *damage-agent* have no obvious lexicalized prototype, so we need a more generic means of identifying the prototypical member of a category. Lakoff [4] suggests that the prototype will occupy a central position in the category's structure, with other members organized in a radial fashion, at a distance from the centre that is inversely proportional to their similarity to the prototype. If we assume that the prototype will be that member that is most evocative of a category, we should first measure the evocation strength of each concept for a given category. This can be done by determining the frequency of occurrence of each concept within the category, and this, in turn, can be estimated by looking to a large corpus to see how each concept is actually employed by language users. Once the most evocative example is found for each category, membership scores can be assigned based on the strength of evocation. The corpus we use must be large, and while reasonably authoritative it must use words both literally and figuratively. For reasons outlined in section 5, we use here as our corpus the complete text of the open-source encyclopaedia Wikipedia [9].

Thus, to estimate the membership level of the word-concept *butcher*|屠夫 in the category *kill-agent*, we first determine the corpus-frequency of the phrase "butcher who kills/killed". In general, for estimating the membership of the concept C in the category *V-agent*, we use the query form "C who|which|that V"; for categories of the form *V-instrument*, we use the query "V with C", and so on. Of course, some verbs are more vague than others, and can have much higher corpus frequencies. We therefore need to normalize raw corpus-frequencies to obtain a truer picture of evocation power. If f_{raw}(V-role:C) denotes the corpus frequency of concept C when considered as a member of the category *V-role*, where V is a verb like "kill" and role is one of *agent*, *instrument*, etc., then the adjusted frequency, a measure of true evocation, is estimated by:

$$f_{adj}(\text{V-role:C}) = ln(f_{raw}(\text{V-role:C})) \times ln(\sum_{x} f_{raw}(\text{V-role:x}))^{-1} \quad (1)$$

Now, the prototype will be that member of a category with the strongest evocation:

$$Prototype(\text{V-role}) = max_c(f_{adj}(\text{V-role:C})) \quad (2)$$

The degree of membership of C in the category V-role is relative to the prototype:

$$Membership(\text{V-role:C}) = f_{adj}(\text{V-role:C}) \times f_{adj}(\text{V-role:}prototype(\text{V-role}))^{-1} \quad (3)$$

This ensures that the prototypical member has a membership score of 1, while all other members of a category will have a score in the range 0... 1. A concept can metaphorically be moved from a category in which it is conventionally a member to any other category in which it is considered to have a non-zero membership score.

4 CLUSTERS AND GROUP-TERMS

For ontological purposes, a category is essentially a cluster of concepts that allows one to conveniently infer similarity – the possession of common properties and shared behaviour – from the simple act of co-categorization. That these clusters often have a heterogeneous roster of members (e.g., as illustrated in Figure 2) is testament both to the prevalence of metaphor and to the necessity of viewing ontological categories as categories of "doing" rather than of "being". Of course, the converse is also true: we can infer the contextual behaviour of a concept from how that concept is explicitly clustered with others. And one common way of signalling the appropriate cluster for a concept is through an evocative group word, like "army", "mob", "tribe" or "coven". For instance, when one uses the phrase "an army of robots", one is conveying a soldier-like perspective on the concept Robot, signalling that in this context, Robot should be viewed more as a attacking agent than as a utensil.

Group terms like "army", "family" and "swarm" are highly suggestive of particular behaviours. For instance, the corpus techniques of section 3 reveal that, in the context of Wikipedia, a "swarm" has two dominant behaviours, *biting* and *attacking*, while an "army" has three, *defeating*, *fighting* and *attacking*. To use the phrase "swarm of X" or "army of X" is to suggest that X also exhibits these behaviours, and furthermore, that X is similar in behaviour to other concepts that comfortably fit these templates. This intuition is easily contextualized, since the relative frequency of these phrases in a context's corpus will reveal the extent to which different concepts belong to different group-based categories.

As a corpus, Wikipedia is biased toward popular culture and genres such as science fiction. This lack of neutrality makes the Wikipedia corpus an excellent example of a context, more so than traditional language corpora. Consider the population of the category *Army-member* as derived from Wikipedia:

mercenary(238), clone(132), soldier(122), volunteer(72), monster(70), robot(63), minion(60), warrior(60), frog(58), knight(50), slave(48), demon(46), clansman(46), monkey(46), crusader(44), gladiator(38), ant(37), lawyer(32), contributor(28), mutant(27), ...

Note the prominent presence of the genre elements "clone", "robot" and "minion", as well as examples like "lawyer" for which "army" has a metaphoric meaning. This grouping suggests that lawyers may be seen, alternately, as mercenaries, warriors and even clones, while the extent to which these comparisons are apt in a particular context is a function of how many different groups can contextually claim both as members. For instance, "lawyer" and "warrior" are used with seven different group terms in the Wikipedia corpus – *society, family, cadre, team, army, class and squad,* while "lawyer" and "mercenary" share just three groupings – *team, army, squad*. Interestingly, the most common group term for "lawyer" in Wikipedia is "huddle" (the phrase "huddle of lawyers" occurs 64 times, twice as often as "army of lawyers"), which suggests that, in this context, lawyers are more likely to be categorized as players than warriors, mercenaries, clones or robots

5 PRELIMINARY EMPIRICAL EVALUATION

The choice of corpus is clearly key to the quality of category-membership statistics that can be derived using the methods of sections 3 and 4. This corpus must be large, it must be representative of language use in general, and it should offer a means of search that is robust in the face of noise. At first blush, then, the world-wide-web seems an ideal candidate: in size it is unmatched, and various APIs are available to access powerful search engines like Google. Unfortunately, such APIs rarely provide enough control over the query or the archive to ensure that noise can be eliminated, since these engines typically perform their own stemming and stop-word elimination, putting truly strict matching beyond our reach. This means that common noun-noun collocations, like "fossil record" and "share issue", are easily confused for infrequent or nonsensical noun-verb collocations like "fossils that record" and "shares that issue".

To ensure strict matching with controlled morphology, we require a local text corpus that we can index and search directly, and even subject to part-of-speech tagging. For this reason we choose the collected text of the open-source encyclopaedia Wikipedia [9], which is available to download in XML form. Wikipedia has several obvious benefits as a text corpus: each document is explicitly tagged with a subject-label, since each article defines a specific headword; documents exist in a rich web of interconnections; and documents strive to be authoritative on their subjects. Consider the range of subjects that are found in Wikipedia for the verb "to infect" (with frequencies shown in parentheses):

virus(46), worm(12), retrovirus(7), strain(6), disease(6), bureaucrat(6), poison(4), ally(4), fungus(4), dust(3), smut(2), bacterium(2), physiologist(2), blood(2), plague(2), war(2), substance(2), germ(1), application(1), species(1)

Now consider the range of verbs that can be used with the subject "virus":

infect(46), attack(11), kill(7), jump(6), eat(4), drive(3), produce(3), destroy(3), spread(3), transform(3), escape(2), steal(1), prove(1), carry(1), freeze(1), arrive(1), control(1)

We see from this snapshot that Wikipedia contains enough diversity to capture the dominant application of each verb, and the dominant behaviour of each subject noun. Furthermore, Wikipedia contains enough diversity to reveal creative uses of these nouns and verbs; this snapshot reveals, for instance, that "smut" can "infect" (2 uses) and that a "virus" can "eat", "escape" and even "steal".

One can ask how well these corpus-derived category structures compare with the hand-crafted category structures of HowNet, since one can reasonably expect human-assigned category memberships to be a gold standard for this task. We find that in 69% of cases, the HowNet-assigned category for a given word-concept is also the dominant corpus-derived category, and that in 76% of cases, a word-concept has a statistical membership in the HowNet-assigned category that is greater than the median membership score for that category.

In fact, these results suggest that HowNet is far from being a gold-standard for category membership. In many cases, the HowNet category name is either poorly named or is dangerously misleading. For instance, the primary sense of the verb "doctor" in English is not "heal" but "fiddle" (as in "to doctor one's r*é*sum*é*"). Likewise, HowNet assigns the name "resume" to the super-category of "repair" and "doctor", when the verb "restore" is more appropriate in English. In many other cases, the HowNet assigned category is only one of several that seem intuitively appropriate. For instance, the word "knight" is assigned the dominant category protect-agent (based on 12 occurrences of the pattern "knight who protects") while HowNet assigns it to the category defend-agent (which is the second-most popular corpus assignment, based on 10 occurrences of "knight who defends"). Viewed from this perspective, the corpus-based and hand-crafted approaches to category assignment are complementary, not conflicting, where each can serve to validate and enrich the other.

6 CONCLUSION

The results of our experiments with Wikipedia are promisingly suggestive about the possibility of contextualizing ontological category structures via corpus-derived statistics. For example, the Wikipedia corpus reveals that the most common verb for the subject noun "vampire" is "hunt" (where the phrase "vampires who hunt" occurs 4 times), indicating that in this pop-culture/fantasy-oriented context, a vampire is to be seen predominantly as a member of the category *hunt-agent*, or hunter. While one is unlikely to find such a categorization in an ontology like WordNet, or even Cyc, this is the most appropriate categorization in this context. Nonetheless, these results are hardly conclusive, for although large, Wikipedia is simply not large enough to provide the diversity of evidence needed to reliably derive a heterogeneous category membership. If a resource like Wikipedia lacks the necessary scale, surely this speaks to the futility of defining a context via a corpus?

We believe the answer to this dilemma lies not in ever-larger corpora (which may be too large to preserve the distinctive biases of a given context), but in the combination of different perspectives offered by the same corpus. We have described two different perspectives in this paper: the perspective of behaviour (captured via verb collocations) described in section 3, and the perspective of clustering (captured via group-word collocations) described in section 4. For instance, we know that Robot is the most representative member of the category *army-agent* in Wikipedia (with 63 examples), while army is itself a highly representative member of the category *attack-agent*. This suggests that Robot should also be a strong member of the category *attack-agent*. While Wikipedia records no uses of the collocation "robot who|which|that attacks", this joint perspective is sufficient evidence to support going to the web for this collocation. That is, the intuition that Robot is an *attack-agent* is consistent with the corpus, and thus the context, so the precise membership score can be determined using the larger context of the web.

Bootstrapping techniques like this should allow us to grow more heterogeneous category structures while respecting the ontological biases of the specific context. Once the deficiencies of relatively small corpora are addressed via such techniques, we expect to be better poised to fully explore the ramifications and opportunities of corpus-trained contextual ontologies.

ACKNOWLEDGEMENTS

We would like to thank Enterprise Ireland for supporting this research through a grant from the Commercialization Fund.

REFERENCES

[1] Dong, Z. and Dong, Q., HowNet and the Computation of Meaning, World Scientific, Singapore, 2006.
[2] Glucksberg, S. and Keysar,B., How Metaphors Work, *Metaphor and Thought (2nd edition)*, A. Ortony (Ed.), Cambridge University Press, 1993.
[3] Hofstadter,D., Fluid Concepts and Creative Analogies, Basic Books, 1995.
[4] Lakoff, G., *Women, Fire and Dangerous Things*, Chicago University Press, 1987.
[5] Lenat, D. and Guha, R. V., Building Large Knowledge-Based Systems, Addison Wesley, 1990.
[6] Miller, G. A., WordNet: A Lexical Database for English, *Communications of the ACM, Vol. 38 No.11*, 1995.
[7] Searle, J., Metaphor, *Metaphor and Thought (2nd edition)*, A. Ortony (Ed.), Cambridge University Press, 1993.
[8] Veale, T., Analogy Generation in HowNet, *The proceedings of IJ-CAI'2005*, the International Joint Conference on Artificial Intelligence, 2005.
[9] Wikipedia open-source encyclopaedia: *www.wikipedia.org.*
[10] F van Harmelen, L Serafini and H Stuckenschmidt, C-OWL: Contextualizing ontologies, *Proceedings of the 2nd International Semantic Web Conference*, 2003.
[11] Harris, Z., *Mathematical Structures of Language*, Wiley. 1968.
[12] Hindle, D., Noun classification from predicate-argument structures, *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 268-275, 1990.
[13] P. Cimiano, A. Hotho, S. Staab., Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of AI Research*, Volume 24: 305-339, 2005.
[14] Hearst, M., Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th International Conference on Computational Linguistics (COLING)*, pp. 539-545, 1992.

Towards A Separation of Pragmatic Knowledge and Contextual Information

Robert Porzel and **Hans-Peter Zorn** and **Berenike Loos**[1] and **Rainer Malaka**[2]

Abstract. In this paper we address the question of how *traditional* approaches to modeling world knowledge, i.e. to model shared conceptualizations of specific domains of interest via formal ontologies, can be enhanced by a pragmatic layer to solve the problem of explicating hitherto implicit information contained in the user's utterances and to further the assistance capabilities of dialog systems and how they can be connected to dedicated analyzers that observe topical contextual information. For this purpose, the notions of *context* and *pragmatics* are introduced as one of the central problems facing applications in artificial intelligence. We will argue that pragmatic inferences are impossible without contextual observations and introduce a model of context-adaptive processing using a combination of formal ontologies and analyzers for various types of context.

1 Introduction

In this paper two fundamental, but notoriously tricky, notions for mobile open-domain multimodal human-computer interface systems, such as SmartWeb [26], are discussed as one of the central problems facing both applications in artificial intelligence as well as in natural language processing. These, often conflated, notions are those of *context* and *pragmatics*. Indeed, in many ways both notions are inseparable from each other if one defines pragmatics to be about the encoding and decoding of meaning, which, as pointed out frequently [4, 28, 21], is always context-dependent. This, therefore, entails that pragmatic inferences (also called *pragmatic analyses* [4]) are impossible without recourse to contextual observations. In this paper, we will argue that the distinction between pragmatic knowledge - which is learned/acquired - and contextual information - which is observed/inferred - is of paramount importance in designing scalable context-adaptive systems, which seek to interact with human users and to collaborate intelligently with them. More specifically, we will focus on the use case of natural language understanding using ontology-based analyses of open-domain user utterances.

As the work presented here is part of a research undertaking that attempts to tie together semantic web technologies, natural language processing and assistance systems in an attempt to develop a mobile multimodal open-domain conversational question answering system , the central idea behind it is to employ ontological knowledge - if available - and revert to statistical processing in the absence thereof. In this paper we will focus on the ontology-based processing pipeline and examine how pragmatic knowledge and contextual infromation - needed to increase the conversational capabilities of dialogue systems - can be modeled and consequently employed. For this we give an overview of the state of the art in Section 2, followed by two motivating examples for distinguishing pragmatic knowledge from contextual information in Section 3. Thereafter, we will describe the ontological infrastructure as found in SmartWeb and our approach for modeling pragmatic knowledge as part of that infrastructure in Section 4. Finally, we will show how we *connected* this knowledge to contextual analyzers in Sections 5 and 6 followed by concluding remarks in Section 7.

2 State of the Art

In general, computational pragmatics can be defined as the attempt to enable artificial systems to encode meaning into a set of surface structures or to decode meaning from such forms In this given sense computational pragmatic resolution is equivalent to *decontextualization* in the sense of McCarthy [17]. While this work will, from now on, focus on the decoding processes it is theoretically quite possible to apply the same techniques to processes of encoding, but will not be the focus of this paper. As we will show herein, there are sound theoretical as well as practical reasons for modularizing and separating pragmatic knowledge, for which we propose an ontological model called PRONTO, from contextual information, which has to integrate numerous non-discrete, noisy and sub-symbolic sensor data in a robust fashion, for which dedicated analyzers and inference mechanisms for combining various observations can be employed.

In general terms, decoding meaning is *understanding*, however, no precise notions of where semantic processing ends and pragmatic processing begins exists, and might never be forthcoming. Various overviews describing the need for context-adaptiveness in natural language processing systems exist [4, 6, 21]. Given the goal of more intuitively usable and more conversational natural language interfaces that can someday be used in real world applications, the handling of pragmatic knowledge - needed for a felicitous decoding of the meaning encoded in user's utterances - is still one of the major challenges for understanding conversational utterances in dialogue systems, since a substantial part of that meaning is contained implicitly in the linguistic surface structures of the utterance, recourse to contextual information is needed for pragmatic analyses. The paramount importance of context for natural language understanding is frequently noted in the literature, albeit few dialogue systems take context explicitly into account and perform a corresponding context-dependent analysis of the given utterances at hand. We follow Porzel and Gurevych [21] and differentiate between four different types of contexts that contribute information relevant to natural language understanding, listed in Table 1. In dialogue systems these knowledge stores are commonly assigned to respective models: the situation model, dialogue model, user model and the domain model, e.g. represented in a formal ontology.

[1] European Media Laboratory, 69118 Heidelberg, Germany, email: firstname.lastname@eml-d.villa-bosch.de
[2] University of Bremen, 28359 Bremen, Germany, email: lastname@informatik.uni-bremen.de

Table 1. Context-types, content and their models

types of context	information observed	context model
situational context	time, place, etc	situation model
discourse context	what has been said	discourse model
interlocutionary context	user/system properties	user/system model
domain context	ontological knowledge	domain model

Recently developed multi-modal dialogue systems [27, 13, 23] equipped with the ability to understand and process natural language utterances from one ore more domains often employ ontologies as a formal, explicit specification of shared conceptualizations of their domains of interest [10]. At the same time the emerging Semantic Web [2] employs such formal conceptualizations to add semantic information to textual and other data available on the Internet. Efforts originating in various W3C and Semantic Web projects brought about several knowledge modeling standards: Resource Description Framework (RDF), DARPA Agent Mark-up Language (DAML), Ontology Interchange Language (OIL) and the Ontology Web Language standards (OWL (Lite, DL, Full)).[3]

Therefore, numerous mobile dialogue systems, such as MATCH, SmartKom or SmartWeb [27, 13, 26], employ ontologies to represent spatial and navigational knowledge; to support car, motorcycle and pedestrian navigation. Existing navigation ontologies [16, 12] describe route mereologies, which do not capture contextual dependencies. The same holds true for other domain ontologies used by the individual system(s), e.g. models of domains such as sports, entertainment and the like. Also, while ontologies commonly model a more or less static world, conceptual and common-sense knowledge [25, 11, 5] based on the standard combinations of frame- and description logics, contextual knowledge is induced in specific *instances* and highly dynamic states of affairs. In natural language processing many ambiguities arise, which can be resolved only by recourse to different contexts, e.g. discourse context has to be taken into account for reference resolution [9], domain context for hypothesis verification [22] or situational context for resolving pragmatic ambiguities [20].

Visible in all systems that are limited to an impoverished contextual analysis and precompilations, was their restrictedness in terms of their understanding capabilities, rendering them unscalable and in the case of more conversational input undeployable. This evidently shows up in the fragility of systems that fail when confronted with imperfect or unanticipated input, usually that also include perfectly unambiguous utterances that stray but a little from a scripted demo dialogue. Human conversations are between partners that share a rich background of pragmatic knowledge (involving topical observations of both more static & more dynamic contexts) without which natural language utterances become ambiguous, vague and incomplete. An interpreter with little contextual awareness and pragmatic reasoning will encounter problems and fail frequently; one which does not fail in unexpected or more complex situations is called *robust*. Several means have been used to increase robustness ranging from rules for grammatical relaxations, automatic acquisition of semantic grammars, automatic spelling correction to on-line lexical acquisition and out-of-vocabulary recognition. These so-called *low-level techniques* [4] have not solved the problem of enabling a system to react felicitously in dynamic contexts and for multiple domains. These techniques fail to assume a pragmatics-based approach where the fact that the user has an intention, communicated via a message, which has to be reconstructed by recourse to the current context, is explicitly taken into account. Therefore, today's systems using *pragmatics-free* ontologies face two options. One is to to restrict themselves to single applications with clearly defined application-specific contexts, e.g. offering single domain services - such as providing information about soccer scores - or guiding only pedestrians - always on foot and always on the shortest path. The other is to force the user to explicate each possible contextual parameter, which means reverting to controlled and restricted processing techniques.

However, if we wish to make use of (or combine) semantically described web services, which offer vast ensembles of tunable parameters, e.g. route, weather, and geo-services, or to employ semantic information extraction applications in a variety of domains, e.g. sports or news, we must provide the means to decode the appropriate meaning based on pragmatic knowledge and context-specific topical information. Moreover, we would like to do so in the least invasive way, i.e. minimizing the amount of information that needs to be obtained by asking the user in order to maximize dialogical efficiency and user satisfaction.In the following we motivate and describe how the ontologies used in the SmartWeb project were adapted to provide a principled approach for encoding pragmatic knowledge.

3 Contextual Information and Pragmatic Knowledge at Play

As mentioned above we apply our model of pragmatic knowledge and context-dependent processing to enhance the conversational understanding and ensuing assistance capabilities of dialog systems. While there exists quite a slippery slope where semantic processing ends and pragmatic assistance begins, we will try to motivate this distinction by means of two sample scenarios employed as running examples throughout this paper.

A question such as *How often was Brazil world champion?* poses a challenge to conversational open-domain dialog systems as the discourse domain of the utterance is not made explicit by the user. Since we regard the modeling of pragmatic knowledge as a major challenge for such systems and - in contrast to controlled systems - want the user to be able to make utterances in any domain of interest without placing the burden of explicating the exact context on him or her, we have to find a systematic and scalable way of modeling:

- that the pragmatic knowledge that a *correct* or felicitous answer to such a question (or many others for that matter) simply depends on what is talked about, and
- that any *intelligent* interlocutor has to know, keep track of or infer what is being talked about.

While these two statements may sound trivial, they are not. For one, the first statement expresses a fundamental bit of pragmatic knowledge that, to the best of our knowledge, has been proposed, implemented and evaluated in dialog systems only by Zorn et al. [15].[4]. This model explicetely and formally expresses such pragmatic knowledge, e.g. a bit that expresses that the *theme* of an utterance - what is new, unknown and asked about - depends on the given *rheme* - what is old, known and has been talked about. In Section 4 we show describe the corresponding ontological framework and in Section 5 how we integrate such knowledge with actual contextual observations, which as expressed in the second statement and can be regarded as an *observational* task assigned to the discourse model.

[3] See www.w3c.org/RDF, www.ontoknowledge.org/oil, www.daml.org, and www.w3.org/2004/OWL for the individual specifications.

[4] Of course, as shown in Section 2 most systems assume an implicitely given domain context or employ various shortcuts to deal with problems of underspecification.

That is to keep track and make inferences about what is being talked about or, in our terminology, to observe the given rheme at hand, which - as all contextual information - can change dynamically and even rapidly.

In a mobile dialog system contextual information is of high significance as a user expects the offer of topical services, while navigating through a dynamically changing environment (e.g. changing precipitation- and temperature levels and or traffic- and road conditions), which makes the adequate inclusion of extra-linguistic knowledge and context-sensitive processing inevitable for the task of felicitous navigational assistance. The necessity to couple extra-linguistic situative with pragmatic knowledge in the domain of spatial navigation has been demonstrated before [20, 14]. Some more obvious examples are given below:

- For instance, a pedestrian might prefer public transportation over walking when it is raining even for smaller distances.
- A motor bicyclist might prefer to use winding country roads over interstate highways when it is warm and sunny, but not, when road conditions are bad.
- A car driver might like to take a spatially longer route if shorter ones are blocked or perilous.

As mentioned above, existing navigation ontologies [16, 10] describe route mereologies, which do not capture contextual dependencies. Given a single application-specific context, e.g. guiding only pedestrians - always on foot and always on the shortest path, we can employ such a *context-free* ontology. However, if we wish to make use of the many tunable parameters offered by today's route planning and navigational systems one must provide the means to determine the right setting depending on the actual situation at hand in the least invasive way, i.e. minimizing the amount of parameters and settings obtained by bothering the user. In the following we motivate our ontological choices and describe the infrastructure employed in our approach to model the needed pragmatic knowledge for solving both sample use cases described above.

4 Pragmatic and other Ontologies in the SmartWeb Project

In order to allow systems such as the SmartWeb prototype [23] to employ a wide range of internal and external ontologies several ontological commitments and choices have to be made. The most relevant for our work are described below.

Foundational & Ground Knowledge: An important aspect in ontology engineering is the choice of a foundational layer, which is used to guarantee harmonious alignment of various independently crafted domain ontologies and their re-usability. The SmartWeb foundational ontology [5] is based on the highly axiomatized Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) It features various extensions called *modules*, e.g. the Ontology of Plans and a module called *Descriptions & Situations* [8]. As the focus of our work lies on an application and elaboration of the latter module, it will be described more closely in the following section. Additional to the foundational ontology, a domain-independent layer is included which consists of a range of branches from the less axiomatic SUMO (Suggested Upper Merged Ontology ontology [18]), which is known for its intuitive and comprehensible structure. Currently, the SmartWeb Integrated Ontology (SWINTO) features, next to the foundation and domain-independent layers, several domain ontologies, i.e. a SportEvent-, a Navigation-, a WebCam-, a Media-, and a Discourse-Ontology.

Pragmatic Descriptions & Situations: The Descriptions & Situations framework is currently the sole ontological framework for representing a variety of reified contexts and states of affairs. In contrast to physical objects or events, the extensions of ontologies by non-physical objects pose a challenge to the ontology engineer. The reason for this lies in the fact that non-physical objects are taken to have meaning only in combination with some other *ground* entity. Accordingly, their logical representation is generally set at the level of theories or models and not at the level of concepts or relations. According to Gangemi and Mika [8] this is not generally true as recent work can address non-physical objects as first-order entities that can change, or that can be manipulated similarly to physical entities. So in many cases relations and axioms modeled and applied for physical entities are also valid for non-physical ones. Therefore, a modeling pattern was devised that connects:

- COURSES OF EVENTS sequenced by PERDURANTS, i.e. processes within the ground ontology, such as QUESTIONING,
- FUNCTIONAL ROLES played by ENDURANTS, i.e. objects within the ground ontology, such as a type of EVENT or BUILDING,
- PARAMETERS valued by REGIONS, i.e. scalar phenomena, such as TEMPERATURES or DOMAINS

For endowing the SmartWeb ontologies with a pragmatic layer, we, therefore, decided to employ the *Descriptions & Situations* (D&S) module and its modeling patterns. The central modeling choice that arises hereby concerns the question of how fine-grained such a description and relation hierarchy should be that links the corresponding courses, roles and parameters to elements of the ground ontology. Hereby the classic trade-off between modeling and axiomatization comes into play, i.e. if a corresponding axiomatization should bear the burden of associating the pragmatically grouped items of the ground (domain) ontologies, e.g. SOCCER DISCOURSE, WORLD CUP and QUESTIONING for describing the pragmatic context of a given question. In either case this elaboration of the *Descriptions & Situations* module extends the notion of deriving an instance (situation) from a description by modeling a more general pattern of pragmatic knowledge.

5 Connecting Pragmatic Knowledge with Contextual Observations

Our context model - used for observing contextual information - is implemented as a module, called Situation and Context Module (SitCoM) within `SmartWeb`'s dialog manager. It interacts with the dialog manager's iHUB middle-ware [24] . The internal communication format in SmartWeb is a RDFS adapted derivative of the EMMA w3c standard called SWEMMA. A SWEMMA document is a collection of instances, the actual interpretation is embedded within instances of a discourse and a special EMMA domain ontology. Within the dialog manager these EMMA documents are stored in an A-box. All dialog manager components access a common A-box per turn, the internal iHUB contains only pointers to the root instance of an interpretation within this A-box. Each dialog component then adds its own interpretation to the EMMA document.

SitCoM receives the semantic interpretation via the iHUB, which has been processed by the modality specific recognizers (e.g. for speech and gesture), parser and discourse model components before. The task for SitCoM is to change the semantic representation in such way that contextual information is semantically represented, as if the user would have done so explicitly. If no pragmatic descriptions are applicable the A-box is not modified and the message is sent back to the iHUB without any changes. For a pragmatic description to be applicable means that any of the ground entities contained in the SWEMMA document have been connected to COURSES OF EVENTS, FUNCTIONAL ROLES or PARAMETERS via the respecitve relations *sequenced by*, *played by* or *valued by*.

If SitCom can apply its pragmatic knowledge it will enhance the semantic representation of the user utterance. This is done either by specializing a concept or inserting missing instances into the interpretation. The necessary information stems from connections established to context providing services or sensors. Currently, we query web services for topical weather and road conditions, establish the user's current position via GPS build into the mobile device and communicate with other components of the system to obtain discourse and temporal information.

As stated above in a mobile dialogue system contextual information is of paramount importance as the user expects the offer of topical services. This alone makes the adequate inclusion of contextual factors intertwined with the corresponding pragmatic knowledge inevitable for the task of navigational assistance.

However, a closer examination shows that in a truly open domain system, such as SmartWeb, virtually every utterance becomes ambiguous in an open-domain context. Looking, again, at the question introduced above, i.e. *How often was Brazil world champion?*, we find that, without knowing the domain at hand, i.e. which type of sport - soccer, beachball or else - is talked about, it is not possible to answer these questions directly. Currently, this problem is handled by either restricting NLU systems to a pre-specified (hard-coded) domain or shifting the pragmatic disambiguation task back to the user, by asking him or her to specify the needed information, thereby producing less efficient and more cumbersome dialogues.

6 Adding Context to the System

Our context model - used for observing contextual information - is implemented as a module, called Situation and Context Module (SitCoM) within `SmartWeb`'s dialog manager. It interacts with the dialog manager's IHUB middle-ware [24] . The internal communication format in SmartWeb is a RDFS adapted derivative of the EMMA w3c standard called SWEMMA. A SWEMMA document is a collection of instances, the actual interpretation is embedded within instances of a discourse and a special EMMA domain ontology. Within the dialog manager these EMMA documents are stored in an A-box. All dialog manager components access a common A-box per turn, the internal IHUB contains only pointers to the root instance of an interpretation within this A-box. Each dialog component then adds its own interpretation to the EMMA document.

SitCoM receives the semantic interpretation via the IHUB, which has been processed by the modality specific recognizers (e.g. for speech and gesture), parser and discourse model components before. The task for SitCoM is to change the semantic representation in such way that contextual information is semantically represented, as if the user would have done so explicitly. If no pragmatic descriptions are applicable the A-box is not modified and the message is sent back to the IHUB without any changes. For a pragmatic description to be applicable means that any of the ground entities contained in the SWEMMA document has been connected to COURSES OF EVENTS, FUNCTIONAL ROLES or PARAMETERS via the respecitve relations *sequenced by*, *played by* or *valued by*. Additional inferencing mechanisms are needed for selecting appropriate descriptions, insertions of appropriate concepts and instances and combinations of observations, which have been proposed and are described in greater detail by Chang et al [3], Porzel et al. [20].

If SitCom can apply its pragmatic knowledge it will enhance the semantic representation of the user utterance. This is done either by specializing a concept or inserting missing instances into the interpretation. The necessary information stems from connections established to context providing services or sensors. Currently, we query web services for topical weather and road conditions, establish the user's current position via GPS build into the mobile device and communicate with other components of the system to obtain discourse and temporal information.

If `SitCoM` can apply its pragmatic knowledge it will enhance the semantic representation of the user utterance. This is done either by specializing a concept or inserting missing instances into the interpretation. The Situation and Context Module (SITCOM) is connected to other dialog processing modules, i.e. Speech Interpretation (SPIN), Fusion and Dialog Engine (FADE), Reaction and Presentation Manager (REAPR), the EMMA Unpacker/Packer that handles communication with the multimodal recognizer and the semantic mediator which manages access to the knowledge access services, within SmartWb's multimodal dialog processing architecture. In the following we will describe the processing steps undertaken by our module.

Collecting Pragmatic Descriptions: The `SitCoM` algorithm performs two passes over the instances contained in the SWEMMA documents found in the iHUB. These instances are part of the ground ontology and are bound via their respective properties to pragmatic description modelled in our pragmatic ontology (PrOnto). This way, the ground entities *evoke* certain description which describe contexts or situations in which the given concept may play a role. In the first pass, all these evoked descriptions are collected and put in an *active descriptions* pool.

Context Sources: The interface to the sensor data is encapsulated into so called context sources. These context sources are identified by a concept from the ground ontology and provide the context information as instance of this concept or a subclass of it. The context information can be a set of instances, in this case, the identifying concept is the anchor instance. Below, we describe a set of sources that are currently analyzed by our module.

- A GPS Receiver connected to the user device delivers current location data to the dialog manager which is passed as *external message* to `SitCoM` by the IHUB in small intervals. The GPS context source uses a web service to resolve the exact address using inverse geocoding. This information is cached and only updated if the location has changed significantly.
- The Weather Service context source polls a Web Service for current weather conditions depending on the current location.
- The Time context source encapsulates time information from the real time clock.
- This context source provides the current domain as recognized by a domain recognizer.

Context Insertion Step: These descriptions are matched against the context information and - if applicable - accordingly specialized. The parameter of the description is used to query the context source. If the resulting context information instance is some subclass of this parameter, the corresponding description-subclass is activated instead.

The last step is another iteration over all instances of the current interpretation. During this pass, all concepts are matched against the description within the active descriptions pool. If a description has been specialized in the previous pass, the ground entities corresponding to this more specific description are specialized as well.

For example: A `Tournament` instances evokes the "SportsTalk" description. This description is about talking about specific domains, e.g. sports. It consists of the functional Role `SportsRhema`, the parameter `SportsThema`. `SportsRhema` is connected to the `Tournament` ground entity and this way the description gets activated. `SportsThema` is linked to the `Domain` ground entity which is covered by the `Domain` context source. This context source returns an instance of `SoccerDomain` which is a subclass of `Domain`. This way a sub description "SoccerTalk", consisting of `SoccerRhema` and `SoccerThema` gets active. During the last step the Tournament instance is changed to a FIFAWorldCup instance to match the more specialized "SoccerTalk" description where the functional role is linked to.

7 Conclusion

In this paper we have argued that an inclusion of pragmatic knowledge is needed to scale context-adaptive systems and that this inclusion can be achieved by means of an ontological model based on an extension of the situations & descriptions framework. Additionally, we have pointed at the need to handle contextual information differently from pragmatic knowledge, as it is quite different in nature and requires other classification, inferencing and reasoning methods, for which ontologies are simply not suitable. As future work, a promising framework, called BayesOWL, originating in the work of Ding [7] constitutes a promising next step towards a better integration of symbolic and probabilistic reasoning. Additionally, the framework proposed by Porzel [19] can be employed to integrate the various contextual observations in probabilistic graphical models while keeping the conditional probability tables from exploding.

Acknowledgements

This work has been partially funded by the German Federal Ministry of Research and Technology (BMBF) as part of the SmartWeb project under Grant 01IMD01E and by the Klaus Tschira Foundation. We would like to thank the referees for their comments which helped improve this paper.

REFERENCES

[1] James Allen, Georga Ferguson, and Amanda Stent, 'An architecture for more realistic conversational system', in *Proceedings of Intelligent User Interfaces*, pp. 1–8, Santa Fe, NM, (2001).

[2] Tim Berners-Lee, James Hendler, and Ora Lassila, 'The semantic web', *Scientific American*, **May**, (2001).

[3] John Bryant, Nancy Chang, Robert Porzel, and Keith Sanders, 'Where is natural lnaguage understanding', in *Proceedings of AMLAP*, (2001).

[4] Harry Bunt, 'Dialogue pragmatics and context specification', in *Computational Pragmatics, Abduction, Belief and Context*, John Benjamins, (2000).

[5] Philipp Cimiano, Andreas Eberhart, Daniel Hitzler, Pascal Oberle, Steffen Staab, and Rudi Studer, 'The smartweb foundational ontology', *SmartWeb Project Report*, (2004).

[6] John H. Connolly, 'Context in the study of human languages and computer programming languages: A comparison', *Modeling and Using Context*, **Springer, LNCS**, 116–128, (2001).

[7] Zhongli Ding, *BayesOWL: A Probabilistic Framework for Semantic Web*, Ph.D. dissertation, University of Maryland, 2005.

[8] Aldo Gangemi and Peter Mika, 'Understanding the semantic web through descriptions and situations', in *Proceedings of the ODBASE Conference*. Springer, (2003).

[9] Peter C. Gordon and Randall Hendrick, 'The representation and processing of coreference in discourse', *Cognitive Science*, **22**(4), 389–424, (1998).

[10] Thomas Gruber, 'A translation approach to portable ontology specifications', *Knowledge Acquisition (5)*, (1993).

[11] Nicola Guarino, 'Some ontological principles for designing upper level lexical resources', in *Proceedings of the 1st International Conference on Language Resources and Evaluation,*, (1998).

[12] Iryna Gurevych, Robert Porzel, Elena Slinko, Norbert Pfleger, Jan Alexandersson, and Stefan Merten, 'Less is more: Using a single knowledge representation in dialogue systems', in *Proceedings of the 3rd IJCAI 2003 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, p. (in press), Acapulco, Mexico, (2003).

[13] Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent, Patrick Ehlen, Marilyn Walker, Steve Whittaker, and Preetam Maloor, 'Match: An architecture for multimodal dialogue systems', in *Proceedings of ACL'02*, pp. 376–383, (2002).

[14] Berenike Loos and Robert Porzel, 'Towards ontology-based pragmatic analysis', in *In Processding of DIALOR'05*, Nancy, France, (2005).

[15] Hans-Peter Zorn Berenike Loos and Robert Porzel, 'Pragmatic knowledge and context-aware proccesing for dialogue systems', in *7th SIGdial Workshop on Discourse and Dialogue Sydney, July 15-16,*, (submitted).

[16] Raphael Malyankar, 'Creating a navigation ontology', in *Workshop on Ontology Management*, Menlo Park, CA, (1999). AAAI Press.

[17] John McCarthy, 'Notes on formalizing contexts', in *Proceedings of the Fifth National Conference on Artificial Intelligence*, eds., Tom Kehler and Stan Rosenschein, pp. 555–560, Los Altos, California, (1986). Morgan Kaufmann.

[18] Ian Niles and Adam Pease, 'Towards a standard upper ontology', in *Workshop on Ontology Management*, eds., Chris Welty and Barry Smith, Ogunquit, Maine, (2001). Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001).

[19] Robert Porzel, 'Baysian networks with expert nodes: An application in computational pragmatics', in *Master Thesis at the International University in Bruchsal*, (2006).

[20] Robert Porzel and Iryna Gurevych, 'Towards context-adaptive utterance interpretation', in *Proceedings of the 3rd SIGdial Workshop*, Philadelphia, USA, 2002, pp. 90–95, (2002).

[21] Robert Porzel and Iryna Gurevych, *Contextual Coherence in Natural Language Processing*, LNAI 2680, Sprigner, Berlin, 2003.

[22] Robert Porzel and Iryna Gurevych, 'Contextual coherence in natural language processing', in *Fourth International Conference on Modeling and Using Context*, eds., P. Blackburn, C. Ghidini, R. Turner, and F. Giunchiglia, Berlin, (2003). Springer (LNAI 2680).

[23] Norbert Reithinger, Simon Bergweiler, Ralf Engel, Gerd Herzog, Norbert Pfleger, Massimo Romanelli, and Daniel Sonntag, 'Look under the hood: Design and development of the first smartweb system demonstrator', in *Proceedings of IMCI*, Trento, Italy, (2005).

[24] Norbert Reithinger and Daniel Sonntag, 'An integration framework for a mobile multimodal dialogue system accessing the semantic web', in *Proceedings of Interspeech 2005*, Lissabon, Portugal, (2005).

[25] Stuart J. Russell and Peter Norvig, *Artificial Intelligence. A Modern Approach*, Prentice Hall, Englewood Cliffs, N.J., 1995.

[26] Wolfgang Wahlster, 'SmartWeb: Mobile applications of the semantic web', in *Proceedings of Informatik*, Ulm, Germany, (2004).

[27] Wolfgang Wahlster, Norbert Reithinger, and Anselm Blocher, 'Smartkom: Multimodal communication with a life-like character', in *Procedings of the 7th Eurospeech.*, pp. 1547–1550, (2001).

[28] Dominique Widdows, *A Mathematical Model of Context*, LNAI 2680, Sprigner, Berlin, 2003.

Enforcing a Semantic Routing Mechanism based on Peer Context Matching

Silvana Castano and **Stefano Montanelli** [1]

Abstract. In this paper we present the main features of the H-LINK semantic routing mechanism we are developing to combine ontology-based peer contexts and ontology matching techniques for providing P2P query forwarding on a real semantic basis. H-LINK defines a semantic overlay network where each edge represents a semantic link between two peers having similar contexts. Semantic links are exploited to address query propagation by identifying the semantic neighbors that can provide relevant knowledge with respect to a given target request.

1 INTRODUCTION

Recent schema-based P2P networks go beyond traditional file-sharing P2P networks, by providing infrastructures where peers can create and share knowledge [1]. In this scenario, peers join the system by providing their own context and need to cooperate by matching their respective context with the aim to discover similar partners and to enforce effective resource sharing. In order to provide scalable infrastructures for peer communications, P2P semantic routing protocols are being proposed with the aim to address query propagation on the basis of the local context of each peer [2, 7, 9, 12, 14]. At the current stage of development, a challenging issue regards the need of advancing the existing semantic routing protocols by combining ontology-based peer contexts and ontology matching techniques for providing query forwarding on a real semantic basis.

In this paper, we present the main features of the H-LINK semantic routing mechanism we are developing in the framework of our HELIOS peer-based system for knowledge sharing and evolution [5]. In HELIOS, the peer context is represented through a *peer ontology* describing the knowledge the peer brings to the network and the knowledge the peer perceives from the network. Peers act as independent agents with their own context (i.e., peer ontology) and interact each other by submitting discovery queries and by replying with relevant knowledge. In the HELIOS framework, the H-MATCH semantic matchmaker has been developed to evaluate the semantic affinity between an incoming discovery query and a peer ontology. On this basis, the H-LINK semantic routing mechanism is designed to exploit the matching knowledge acquired from the discovery process. The matching knowledge becomes *network knowledge* in the peer ontology, and it is exploited to provide a semantic overlay network where peers having similar contexts are interlinked as *semantic neighbors*. This way, as a peer learns about the network contents through discovery queries, also its network knowledge is gradually evolved to reflect its newly acquired semantic neighbors.

Example of knowledge discovery in HELIOS. Considering the scenario of Figure 1, we suppose that peer A is interested in discovering peers capable of providing resources semantically related to the publishing domain. To this end, peer A composes and submits to the system a discovery query Q1 containing the target concepts of interest Publication and Book with the properties year and author, respectively. Moreover, Book is specified as a subclass of Publication. Receiving the query Q1, the peer (i.e., peer B, peer C, and peer D) uses the H-MATCH semantic matchmaker to compare the query target with its own peer ontology, with the aim to identify whether there are concepts matching the target request. According to their matching results, peer B and peer D send back to the requesting peer A a ranked list of concepts found to be semantically related to the target, and, for each entry, the corresponding semantic affinity value SA. In particular, peer B replies with the Volume matching concept as $SA(Book, Volume) = 0.82$, while peer D sends back two matching concepts, namely Newspaper and Magazine, with $SA(Publication, Newspaper) = 0.67$ and $SA(Book, Magazine) = 0.539$. On the other hand, peer C does not reply to peer A as no matching concepts are identified. The query replies represent the discovered knowledge of peer A that can be exploited to decide whether to further interact with the answering peers in order to access their relevant resources for data sharing. Before H-LINK, the discovery process relied on the conventional P2P infrastructure and associated routing protocols for addressing query propagation in the network. In H-LINK, we show how the discovered knowledge can be further exploited for semantic routing purposes by enforcing query forwarding according to peer context similarities.

2 ONTOLOGY MATCHING WITH H-MATCH

H-MATCH performs ontology matching at different levels of depth by deploying four different *matching models* spanning from surface to intensive matching, with the goal of providing a wide spectrum of metrics suited for dealing with many different matching scenarios that can be encountered in comparing concept descriptions of real ontologies. H-MATCH takes two ontologies as input and returns the mappings that identify corresponding concepts in the two ontologies, namely the concepts with the same or the closest intended meaning. H-MATCH mappings are established after an analysis of the similarity of the concepts in the compared ontologies. In H-MATCH we perform similarity analysis through affinity metrics to determine a measure of semantic affinity in the range $[0, 1]$. A threshold-based mechanism is enforced to set the minimum level of semantic affinity required to consider two concepts as matching concepts. Given two

[1] Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano. Via Comelico 39, 20135 Milano, Italy. Email:{castano,montanelli}@dico.unimi.it

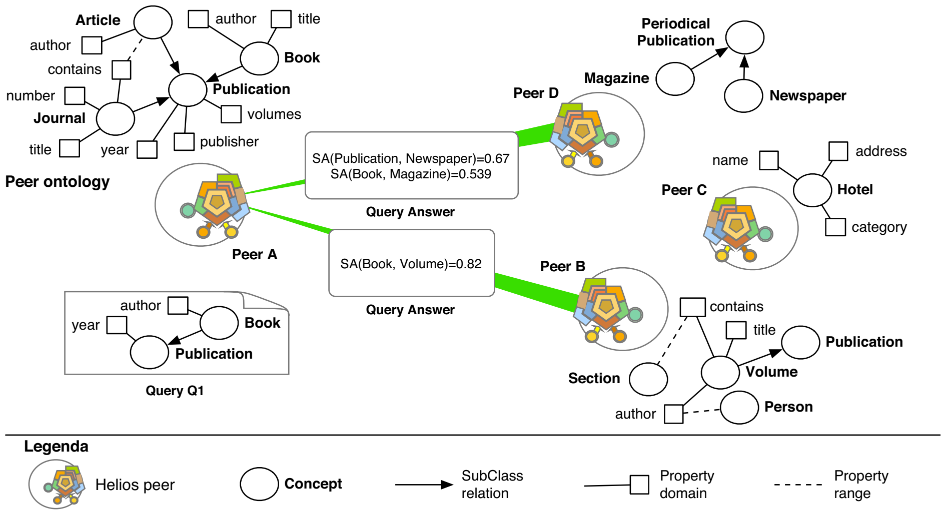

Figure 1. Example of knowledge discovery in HELIOS

concepts c and c', H-MATCH calculates a semantic affinity value $SA(c, c')$ as the linear combination of a linguistic affinity value $LA(c, c')$ and a contextual affinity value $CA(c, c')$. The linguistic affinity function of H-MATCH provides a measure of similarity between two ontology concepts c and c' computed on the basis of their linguistic features (i.e., concept names). For the linguistic affinity evaluation, H-MATCH relies on a thesaurus of terms and terminological relationships automatically extracted from the WordNet lexical system. The contextual affinity function of H-MATCH provides a measure of similarity by taking into account the contextual features of the ontology concepts c and c'. The context of a concept can include properties, semantic relations with other concepts, and property values. The context can be differently composed to consider different levels of semantic complexity, and four matching models, namely, *surface*, *shallow*, *deep*, and *intensive*, are defined to this end. In the surface matching, only the linguistic affinity between the concept names of c and c' is considered to determine concept similarity. In the shallow, deep, and intensive matching, also contextual affinity is taken into account to determine concept similarity. In particular, the shallow matching computes the contextual affinity by considering the context of c and c' as composed only by their properties. Deep and intensive matching extend the depth of concept context for the contextual affinity evaluation of c and c', by considering also semantic relations with other concepts (deep matching model) as well as property values (intensive matching model), respectively. The comprehensive semantic affinity $SA(c, c')$ is evaluated as the weighted sum of the Linguistic Affinity value and the Contextual Affinity value, that is:

$$SA(c, c') = W_{LA} \cdot LA(c, c') + (1 - W_{LA}) \cdot CA(c, c') \qquad (1)$$

where W_{LA} is a weight expressing the relevance to be given for the linguistic affinity in the semantic affinity evaluation process.

H-MATCH has been extensively tested on several real ontology matching cases in order to evaluate the matching models with respect to performance and quality of results [4]. By analyzing the obtained results, we note that the most accurate and precise results are achieved with the deep and intensive matching models provided that the ontology descriptions are detailed enough. On the other side, we note that the best performance in terms of computation time are achieved with the surface and shallow matching models. For semantic routing purposes, the computation time of the semantic affinity evaluation is a crucial factor and needs to be performed as fastest as possible in order to avoid bottlenecks. To this end, possible lacks in matching precision and accuracy can be admitted in turn of rapid response time during the semantic affinity evaluation. For this reason, the shallow matching model is selected to work with H-LINK for identifying the semantic neighbors that have the highest chance to provide relevant knowledge with respect to a given query (see Section 4). A detailed description of H-MATCH and related matching models is provided in [4]. We note that H-MATCH can be suitably adopted to enforce semantic routing functionalities by relying on its flexible matching models that allow to dynamically configure the tradeoff between performance and accuracy according to the requirements of the considered matching scenario. Provided that a dynamic and flexible configuration is supported, other existing matching tools can however be used to enforce the H-LINK semantic routing mechanism in turn of H-MATCH [11]. In the remainder of the paper, we focus on the use of H-MATCH for semantic routing in H-LINK.

3 PEER ONTOLOGY ARCHITECTURE

The context of a HELIOS peer is described through a peer ontology that is organized in a two-layer architecture where the upper layer represents the *content knowledge* and the lower layer represents the *network knowledge* of the peer, respectively. The content knowledge layer describes the knowledge the peer brings to the network that is described as a graph of concepts, properties, and semantic relations [2]. The network knowledge layer describes the knowledge that the peer has of the semantic neighbors it has interacted with. With reference to the discovery example in Figure 1, when peer A receives a reply

[2] For the sake of internal representation of ontology specification languages, and in particular for Semantic Web languages like OWL, we rely on a reference model, called H-MODEL, that provides a graph-based representation of peer ontologies. For further details on H-MODEL, the reader can refer to [5].

from peer B and peer D as an answer to the discovery query Q1, it stores in the network knowledge layer a description of peer B and peer D. A peer description is given in the form of *network concept*, characterized by a set of properties describing the network features of the peer (e.g., IP address, bandwidth). A *location relation* is defined to connect a network concept nc with a concept c in the content knowledge layer. The location relation is labeled with a *confidence* annotation cf that keeps track of the discovered semantic affinity between c and the peer ontology of the peer represented with nc. The cf value corresponds to the semantic affinity value SA returned by the peer nc in its query answer. A new location relation is defined for each matching concept returned in the query answer. A comprehensive *expertise* measure is associated with a network concept nc and it is computed as the average mean of the confidence values associated with all the location relations connected with nc.

As an example, in Figure 2, we consider a portion of the peer ontology of the peer A after the knowledge discovery process described in Figure 1. In this example, peer B and peer D have

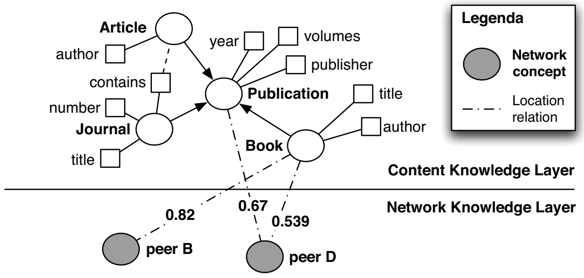

Figure 2. A portion of the peer ontology of peer A

answered to query Q1, then the corresponding network concepts are defined in the network knowledge layer. According to the query reply of peer B, a new location relation with a confidence value of 0.82 is defined to connect the peer B network concept with the Book concept in the content knowledge layer. As two matching concepts are returned in the query answer of peer D (i.e., Newspaper, Magazine), two location relations are defined by connecting the peer D network concept with the concepts Publication and Book in the content knowledge layer, and by setting a confidence value of 0.67 and 0.539, respectively. As a consequence, the expertise measures associated with peer B and peer D are 0.82 and 0.605, respectively.

Considerations. The confidence value associated with a location relation between c and nc is updated when a new semantic affinity value with c is returned by nc in reply to a discovery query. As proposed in [9], the confidence value cf associated with a given location relation between c and nc can be periodically updated by observing the ratio between the number of relevant replies provided by nc and the number of queries sent to nc with a target concept related to c. When the ratio has low values, cf can be decreased to denote that the original confidence (i.e., semantic affinity) is no more actual. In such way, only confirmed location relations are maintained in the peer ontology, while unreliable confidence values are gradually reduced and finally dropped. Furthermore, a number of information can be combined with the confidence measures for providing a more accurate evaluation of the network concept expertise and thus, of the associated semantic neighbor. For instance, a trust mechanism can be adopted to maintain reputation information about the semantic neighbors stored in the network knowledge layer [13]. Moreover, information regarding the network reliability of the semantic neighbors, such as connection stability and granted bandwidth, can be considered for expertise computation [2]. Confidence and expertise measures are exploited by H-LINK for addressing query routing on a semantic basis.

4 THE H-LINK SEMANTIC ROUTING MECHANISM

The H-LINK semantic routing mechanism is based on the idea of exploiting the network knowledge layer of a peer ontology by using the H-MATCH semantic matchmaker for providing query routing support according to semantic neighbor contents.

We consider a query q with a target concept tc [3]. Two different roles can be distinguished for a given peer p:

- *Requesting peer*. Peer p needs to submit to the network a query q in order to identify relevant partners for subsequent resource sharing. To this end, peer p invokes H-MATCH to compare the target concept tc against the content knowledge layer of its peer ontology $\mathcal{O}$. A list $MCL = \{\langle c_1, SA(tc, c_1)\rangle \ldots \langle c_n, SA(tc, c_n)\rangle\}$ of matching concepts $c_1 \ldots c_n \in \mathcal{O}$ and corresponding semantic affinity values $SA(tc, c_1) \ldots SA(tc, c_n)$ is returned as a result. Peer p sets the *number of credits* N_{cr} to distribute to the query recipients in order to define the number of replies that peer p wish to receive as answers to the query q. Therefore, H-LINK is invoked by passing the list MCL to select the semantic neighbors for query q submission.
- *Receiving peer*. When a peer p receives a query q together with the number of credits nc from a requesting peer r, it needs to evaluate whether matching concepts can be provided back to peer r. To this end, H-MATCH is invoked by peer p and the list MCL of matching concepts is still produced as a results. If $MCL \neq \emptyset$, the peer p sends MCL back to peer r by consuming one credit, otherwise no reply is sent back to peer r and all the received credits are still available for forwarding. If at least one credit is available, H-LINK is invoked by peer p to select the semantic neighbors for query q forwarding; otherwise the propagation mechanism stops.

H-LINK invocation. H-LINK is invoked for both query submission/forwarding provided that at least one credit is still available. Three main steps define H-LINK: *selection of semantic neighbors*; *ranking of semantic neighbors*; *distribution of credits*.

1- Selection of semantic neighbors. The network knowledge layer of the peer ontology is accessed to select the network concepts, together with the associated confidence values, that are connected to the concepts in MCL through a location relation. A list SNL of semantic neighbors is returned as a result. A semantic neighbor $sn \in SNL$ is described in the form $sn = \langle nc, \{c_1, cf_1 \ldots c_m, cf_m\}\rangle$, where nc is the network concept featuring sn, while $c_1 \ldots c_m \in MCL$ are the concepts of MCL connected to nc through a location relation, and $\{cf_1 \ldots cf_m\}$ the corresponding confidence values.

2- Ranking of semantic neighbors. The semantic neighbors in SNL are ranked with respect to their relevance for the query target tc. To this end, the harmonic mean is used to combine the

[3] For the sake of clarity, we consider the case of a single target concept in the query. The H-LINK semantic routing mechanism can be easily extended to consider the case of multiple target concepts.

confidence values associated with the semantic neighbors in SNL and the semantic affinity values in MCL. Given a semantic neighbor $sn \in SNL$, the ranking value r_{sn} corresponds to the following formula:

$$r_{sn} = \frac{\sum_{i=1}^{m} \frac{2 \cdot cf_i \cdot SA(tc, c_i)}{cf_i + SA(tc, c_i)}}{m} \quad (2)$$

Finally, a ranked list $RSNL$ of semantic neighbors with the corresponding ranking value is returned as a result. A threshold mechanism can be used to rule out the semantic neighbors with a ranking value lower than a predefined threshold t.

3- Distribution of credits. The semantic neighbors in $RSNL$ determine the recipients of the query q. Available credits A_{cr} are proportionally distributed to the semantic neighbors in $RSNL$ according to their ranking value. Then, the number of credits nc_{sn} assigned to the semantic neighbor $sn \in RSNL$ is computed as follows:

$$nc_{sn} = \lfloor \frac{A_{cr}}{\sum_{\forall sn_i \in RSNL} r_{sn_i}} \cdot r_{sn} \rfloor \quad (3)$$

We note that if H-LINK is invoked with $MCL = \emptyset$, selection and ranking of semantic neighbors are not performed and credits are proportionally distributed according to the expertise measure of the network concepts in the network knowledge layer.

Example. As an example of H-LINK semantic routing, we consider the peer B of Figure 1. Peer B intends to submit to the system the query Q2 described in Figure 3(a) with total number of credits to distribute $N_{cr} = 5$. The peer B uses H-MATCH to compare the query

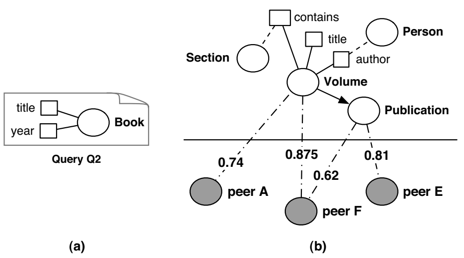

Figure 3. (a) The Query Q2 example and (b) a portion of the peer B ontology

Q2 against its peer ontology (see Figure 3(b)). As a result, the following semantic affinity values are returned by H-MATCH:

SA(Book,Volume)=0.79
SA(Book,Publication)=0.49

By invoking H-LINK, we find that:

MCL={⟨ Volume,0.79 ⟩, ⟨ Publication,0.49 ⟩}
SNL={⟨ peer A,{Volume,0.74}⟩, ⟨ peer E,{Publication,0.81}⟩,
⟨ peer F,{Volume,0.875,Publication,0.62}⟩}

On the basis of such results, H-LINK computes the ranking of the semantic neighbors in SNL and assigns the corresponding number of credits, as shown in Table 1. The query Q2 is then submitted to the selected semantic neighbors together with the assigned number of credits. As shown in the routing schema of Figure 4, peer A receives the query, consumes one credit for replying to peer B, and forwards the query Q2 to peer D by assigning the last remaining credit. Peer E consumes the unique credit received and soon stops the forwarding process, while the peer F forwards all the received credits to peer G as no reply is sent back to peer B.

Table 1. Example of semantic neighbor ranking and credit distribution

Semantic neighbor	Ranking value	Assigned credits
peer A	0.764	2
peer E	0.611	1
peer F	0.689	2

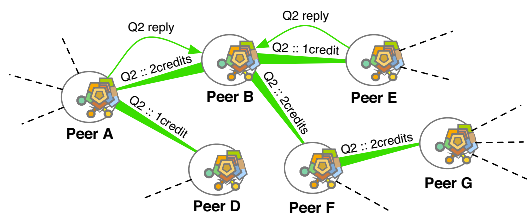

Figure 4. The H-LINK routing schema for query Q2

Considerations. A possible side effect of the H-LINK mechanism is due to the fact that credits are distributed on the basis of the knowledge discovered during past interactions. This means that the knowledge of new peers joining the system is hardly discovered and it is not considered for semantic neighbor selection. H-LINK deals with this by introducing a perturbation during the credit distribution phase. As proposed in [14], a small set of random peers is picked and it receives a percentage of the credits available for distribution. As a result, a larger part of the network is explored with the aim to discover additional knowledge and to include new peers in the semantic routing process.

5 RELATED WORK

Semantic query routing techniques are required to improve effectiveness and scalability of current discovery and search processes for resource sharing in P2P systems. In this direction, the notion of P2P Semantic Link Network is introduced in [15] to emphasize the need of typed semantic links specifying semantic relationships between peers in order to maintain information about nodes with similar contents. Each peer defines its own XML Schema (source schema) describing the contents to share and adopts SOAP-based messages to communicate with the other members of the network. As a difference with location relations in H-LINK, semantic links are exploited with cycle analysis and functional dependency analysis in order to select the query recipients according to the types of the semantic links as well as to the similarity between elements and structures of peer schemas. We note that semantic links need to be actively updated, while location relations are automatically maintained in H-LINK by relying on conventional discovery processes. In [12], the REMINDIN' multi-step query propagation mechanism is described to enforce selected propagation of queries by observing which queries are successfully answered by other peers, by storing these observations, and by subsequently using this information for peer selection. A similar approach is presented in [14] where the Intelligent Search Mechanism (ISM) is introduced to provide an efficient and scalable solution for improving the information retrieval problem in P2P systems. Each ISM peer is composed of four basic elements: i) the *profiling structure* that is used to store the most recent replies of each known peer, ii) the *query similarity function* that is used to identify the similarity between different search queries, iii) the *RelevanceRank algorithm* which exploits the profiling structure to select the peers that can provide relevant answers with respect to a given query, and iv) the *search mechanism* that is used to send the query to the selected peers. As another example of P2P semantic routing approach, the NeuroGrid adaptive decentralized search system is proposed in [9]. In such work, semantic routing is intended as content-based query forwarding, and a learning mechanism is defined to dynamically adjust the relevance of known peers for each query. In NeuroGrid, each node maintains a knowledge base that contains associations between keywords and other nodes. Queries are then forwarded to the nodes that may store matching documents according to the actual knowledge base. In [2], the *Seers* search infrastructure is presented. In Seers, each shared resource is described through a XML *meta-document* and a *matching policy* is used to define how to evaluate the similarity between resources and queries and to assign scores. Scores are then exploited to select the most relevant documents and to rank neighbors for query forwarding. In recent work, ontology-based frameworks are also being proposed to address the lack of semantics in actual P2P routing algorithms. A RDF-based semantic routing architecture is presented in [10]. Nodes are clustered in structured trees according to their interests and intra-/inter-cluster routing algorithms are defined for providing a scalable query forwarding mechanism. In [7], peers advertise their experience in the P2P network according to a shared common ontology. Based on the semantic similarity between a query and the expertise of other nodes, a peer can select appropriate peers for query forwarding.

Original contribution of H-LINK. With respect to the above approaches, we observe that current content-based P2P query propagation algorithms are essentially based on statistical observations and exploit, in some cases, a shared ontology, often mainly a taxonomy. In order to evaluate the similarity between a target query and resources, keyword-based strategies and basic matching techniques (e.g., string matching) are actually supported. The main contribution of H-LINK is related to the use of independent ontologies, rather than a single shared ontology, and to the use of ontology matching techniques to build a network knowledge layer reflecting the gradual learning of semantic neighbors. A further contribution of our approach regards the fact that H-LINK is capable of addressing emergent semantics requirements, by extending current techniques to work in multi-ontology contexts and thus releasing the constraint of having an initial common shared knowledge.

6 CONCLUDING REMARKS AND FUTURE WORK

In this paper, we have presented the H-LINK mechanism we are developing for matching-based semantic routing in P2P systems. Preliminary experimentations show that the H-LINK approach is effective. Our future work will be focused on the extensive experimentation of H-LINK by means of simulation techniques with the aim to assess the real scalability of the proposed approach. Furthermore, we plan to i) investigate the opportunity to refine the credit distribution procedure by considering the recommendation adjustment techniques developed in the field of document retrieval in distributed environments [8], and ii) compare H-LINK with other existing P2P routing approaches in order to evaluate the performance for what concern generated traffic and single peer workload. We will also investigate the opportunity to use flexible ontology evolution techniques for extending the peer ontology with the new concepts that are mostly queried in the network [3], thus improving also peer routing capabilities. Finally, we note that the network concepts keep track of peer context similarities. In this respect, the network knowledge can be exploited for the formation of emergent communities of peers on the basis of their common perspective and context. Some initial results on this topic are presented in [6].

ACKNOWLEDGEMENTS

This paper has been partially funded by the ESTEEM (Emergent Semantics and cooperaTion in multi-knowledgE EnvironMents) PRIN project and by the WEB-MINDS (Wide-scalE, Broadband, MIddleware for Network Distributed Services) FIRB Project funded by the Italian Ministry of Education, University, and Research.

REFERENCES

[1] S. Androutsellis-Theotokis and D. Spinellis, ‘A survey of peer-to-peer content distribution technologies’, *ACM Computing Surveys*, **36**(4), 335–371, (2004).

[2] N.T. Borch and N.K. Vognild, ‘Searching Variably Connected Networks’, in *Proc. of the Int. Conference on Pervasive Computing and Communications (PCC-04)*, Las Vegas, Nevada, USA, (2004).

[3] S. Castano, A. Ferrara, and S. Montanelli, ‘Evolving Open and Independent Ontologies’, *International Journal of Metadata, Semantics and Ontologies*, (2006). To Appear.

[4] S. Castano, A. Ferrara, and S. Montanelli, ‘Matching Ontologies in Open Networked Systems: Techniques and Applications’, *Journal on Data Semantics (JoDS)*, **V**, (2006).

[5] S. Castano, A. Ferrara, and S. Montanelli, *Web Semantics and Ontology*, chapter Dynamic Knowledge Discovery in Open, Distributed and Multi-Ontology Systems: Techniques and Applications, Idea Group, 2006.

[6] S. Castano and S. Montanelli, ‘Semantic Self-Formation of Communities of Peers’, in *Proc. of the ESWC Workshop on Ontologies in Peer-to-Peer Communities*, Heraklion, Greece, (2005).

[7] P. Haase, R. Siebes, and F. van Harmelen, ‘Peer Selection in Peer-to-Peer Networks with Semantic Topologies’, in *Proc. of the Int. Conference on Semantics of a Networked World*, Paris, France, (2004).

[8] M.N. Huhns, U. Mukhopadhyay, L.M. Stephens, and R.D. Bonnell, ‘An Intelligent System for Document Retrieval in Distributed Office Environments’, *Journal of the American Society for Information Science*, **37**(3), 123–135, (1986).

[9] S. Joseph, ‘NeuroGrid: Semantically Routing Queries in Peer-to-Peer Networks’, in *Proc. of the Int. Workshop on Peer-to-Peer Computing (co-located with Networking 2002)*, Pisa, Italy, (2002).

[10] J. Li and S. Vuong, ‘A Scalable Semantic Routing Architecture for Grid Resource Discovery’, in *Proc. of the 11th Int. Conference on Parallel and Distributed Systems*, pp. 29–35, Fukuoka, Japan, (2005).

[11] P. Shvaiko. Ontology Matching Projects, 2006. http://www.ontologymatching.org/projects.html.

[12] S. Staab, C. Tempich, and A. Wranik, ‘REMINDIN’: Semantic Query Routing in Peer-to-Peer Networks based on Social Metaphors’, in *Proc. of the 13th Int. conference on World Wide Web (WWW 2004)*, New York, NY, USA, (2004).

[13] L. Xiong and L. Liu, ‘PeerTrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities’, *IEEE Transactions on Knowledge and Data Engineering*, **16**(7), 843–857, (2004).

[14] D. Zeinalipour-Yazti, V. Kalogeraki, and D. Gunopulos, ‘Exploiting Locality for Scalable Information Retrieval in Peer-to-Peer Networks’, *Information Systems*, **30**(4), 277–298, (2005).

[15] H. Zhuge, J. Liu, L. Feng, and C. He, ‘Semantic-Based Query Routing and Heterogeneous Data Integration in Peer-to-Peer Semantic Link Networks’, in *Proc. of the Int. Conference on Semantics of a Networked World*, pp. 91–107, Paris, France, (2004).

Classification-based Situational Reasoning for Task-oriented Mobile Service Recommendation

Marko Luther,[1] Yusuke Fukazawa,[2] Bertrand Souville,[1] Kunihiro Fujii[2]
Takefumi Naganuma,[2] Matthias Wagner,[1] Shoji Kurakake[2]

Abstract. We study the case of integrating situational reasoning into a mobile service recommendation system. Since mobile Internet services are rapidly proliferating, finding and using appropriate services requires profound service descriptions. As a consequence, for average mobile users it is nowadays virtually impossible to find the most appropriate service among the many offered. To overcome these difficulties, task navigation systems have been proposed to guide users towards best-fitting services. Our goal is to improve the user experience of such task navigation systems by adding context-awareness (i.e., to optimize service navigation by taking the user's situation into account). In this paper we propose the integration of a situational reasoning engine that applies classification-based inference to context elements, gathered from multiple sources and represented using ontologies. The extended task navigator enables the delivery of situation-aware recommendations in a proactive way. Initial experiments with the extended system indicate a considerable improvement of the navigator's usability.

1 Introduction

Within the growing market for mobile Internet, NTT DoCoMo is today providing services to over 50 million mobile phone subscribers in Japan. The majority of these users enjoy widely diverse contents such as entertainment services (ring-tone downloads, games, etc.), transaction services (money transfer, airline reservation, etc.) and information services (weather forecast, maps and local information, etc.) through DoCoMo's high-speed 3G mobile network. Already today, the number of commercial i-mode sites – DoCoMo's brand of mobile Internet services – ranges in the region of many tenth of thousand. With 4G networks at the horizon that promise still substantially higher bandwidth for data transmissions, the market for services with rich content is expected to expand further.

Key to support such growth is the availability of intelligent service platforms that mediate between services and users by observing the users' activity. These platforms have to assist the user in selecting the most appropriate service from the fast growing service pool to support their real world activities, anytime and anywhere.

Our previously developed task-based service retrieval system for the non-expert mobile user makes it easy to retrieve appropriate services for tackling the users challenges in managing his or her everyday life [25]. The term task refers here to "what the user wants to do" as an expression of the users current activity. Furthermore, the system features a task knowledge base, which contains semantic descriptions of potential activities and links to corresponding services that may be helpful. Although this system enables effective service retrieval, it behaves passive in requiring a users initial input to trigger the problem solving process.

In this paper we propose a proactive extension of our basic system that suggests tasks and services actively, without the need for initial user input. This is achieved by the integration of a situation engine and a situation-based task filter, meant to expose only those tasks that are relevant for a user in a given situation. Taking the user's situation into account avoids the necessity of an initial task query. This leads to a considerable improvement of the navigator's usability, especially for non-expert users who are often not willing to input queries.

The abstract characterization of a user's situation is computed by inference mechanisms on several pieces of context information gathered from multiple context sources [20]. We formulate high-level qualitative context elements in the Web Ontology Language (OWL) [22] and concrete situations as instances within the assertional component (Abox) of a situation ontology. To profit from sound, complete and high-performance classifiers such as FaCT++ [31], Pellet [30] and Racer [12], we restrict ourselves to the OWL DL fragment of OWL. To separate concerns we assume that probabilistic aspects of context representation and reasoning are dealt with at lower representation levels applying bayesian networks or fuzzy logics.

The rest of this paper is organized as follows. After discussing related work in the field of ontology-based context reasoning in the next section, we introduce our task-based service navigator application together with some usage scenarios in Section 3. The overall system architecture that underlies the application is presented in Section 4 and the details on our approach to context representation and classification-based reasoning are given in Section 5. In the closing section we report on our experiences gained from this development.

2 Related Work

Several projects consider the use of ontologies as a key requirement for building context-aware applications. Closely related to our approach is the work done in the CALI project [16] as it explores the use of Description Logics (DL) [1] and the associated inferencing. To overcome the limitations of pure DL-based reasoning, a hybrid approach is proposed. However, our earlier experiments [24] indicate that the suggested loose coupling of a DL reasoner with an external generic rule engine leads to serve performance problems. To achieve completeness both reasoners have to be applied successively until no new facts have been derived. Furthermore, it remains unclear how consistency can be guaranteed taking both the knowledge base and the rule base into account.

[1] DoCoMo Euro-Labs, Landsbergerstr. 312, 80687 Munich, Germany
{luther,souville,wagner}@docomolab-euro.com
[2] NTT DoCoMo Inc., 3-5 Hikari-no-oka, Yokusuka, Kanagawa, 239-8536 Japan
{y-fukazawa,naganuma,kurakake}@netlab.nttdocomo.co.jp

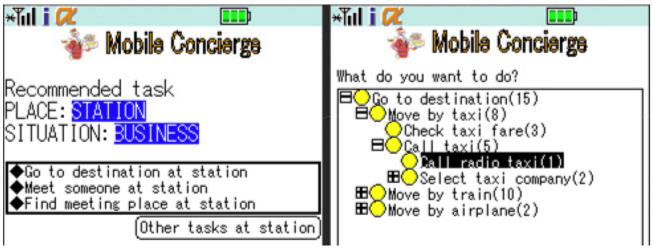

Figure 1. Situation-aware Service Recommender

Other approaches such as CONON [32] and SOUPA/CoBra [4] solely rely on rule-based reasoning which cannot be complete for OWL (not even for OWL Lite [5]) and easily leads to undecidability, as generic rules can be used to simulate role value maps [11].

CONON is an OWL DL encoded upper-context ontology for pervasive computing applications defining almost 200 concepts. Rule-reasoning is used to derive high-level context information and to check its consistency. To cope with the observed delay of several seconds caused by the reasoning process, complex reasoning tasks are computed offline. However, this approach is not feasible in our dynamic setup.

SOUPA, another OWL DL ontology designed for ubiquitous applications, is about the same size as the CONON ontology. Its extension CoBra-Ont is used by a context broker architecture to realize a scenario where people on a university campus come together for a meeting. To limit the reasoning overhead caused by importing standard ontologies, single concepts are mapped to foreign ontology terms. Still, the SOUPA ontology is of a rather high-complexity corresponding $\mathcal{SHOIF}(D)$, because it contains nominals.

An interesting approach to speed up the rule-based inferencing on complex ontologies is to determine relevant contexts required to answer queries using the query-tree method [17]. It remains to be seen how this method extends to our classification-based approach.

3 Situation-aware Service Recommendation

We build on a task-oriented service navigation system [25] that supports the user in finding appropriate services by querying a rich task ontology that represents common sense knowledge about typical complex tasks.

The usage of this basic task navigator is as follows. After having specified a task-oriented query such as “go to theme park” a list of tasks that match this query is sent to the mobile device. Now the user can select the most appropriate task and a corresponding detailed task-model is displayed accordingly. In a final step, associated services can be invoked by establishing an Internet connection to the actual i-mode services.

Figure 1 shows the user interface of the situation-aware variant of the basic service recommender. To explain its functionality, let us assume the following situation.

Situation 1: Important Business Meeting at Tokyo Station
Two travellers, Dawson Campbell and his boss Fiona Davidson, arrive on a Friday morning at the Tokyo main station. Gordon Green, a project partner, is already waiting for them at the platform. The group is looking for a quick transfer to the airport.

Figure 2. Felica Device

To detect the user’s location we further assume that the cell phones of Dawson, Fiona and Gordon are equipped with Felica[3] contact-less RFID tags, enabling a two-way communication with Sonys Felica Reader-Writer devices. Whenever a user puts his phone close to a Felica Reader-Writer device (e.g., to make a mobile payment at a train gate) the recommender application retrieves the corresponding location information as a semantic description of this place (cf. Figure 2). Since Sony and NTT DoCoMo just started to deploy their mobile Suica[4] system for JR East at all stations in the Tokyo region, this assumption is not a fiction but reality.

After having passed the gate at Tokyo station, Dawson’s phone displays a basic list of tasks, associated with the concept *Station*. This list may include entries such as “Prepare to ride a train”, “Buy souvenirs”, “Meet someone” etc. While displaying this task-list, Dawson’s phone connects to the situational reasoning engine and updates his location to *Tokyo station*.

Before having passed the gate, no tasks are shown on Fiona’s phone. Once her location has been detected, a connection to the reasoning engine is established and her current location is updated.

As a result, the situation reasoner infers that Dawson Campbell and Fiona Davidson are travelling together, based on their proximity at the station. In addition, a lookup in the knowledge base reveals that Dawson and Fiona are colleagues and that the scene takes place at a weekdays afternoon.

Because Dawson is located at a *public place* during *office hours* together with *colleagues*, his situation is classified as a *business situation*. His phone shows the inferred situation together with a corresponding list of filtered tasks (shown on the left part of Figure 1). To further specify his needs, Dawson may select one of the recommended tasks (“go to destination” in this case) and finally invoke an associated service (as shown on the right part of Figure 1).

Let us assume another situation taking place at the same location.

Situation 2: Private Meeting at Tokyo Station
Dawson Campbell arrives on a Saturday around noon at the Tokyo main station where Mark Buchanan, his father in law, is awaiting him. They plan to shop for a birthday present for Dawson’s wife.

This situation is classified as *private family meeting*, because it takes place during *leisure hours* and only *relatives* are in the proximity. In this case, the situation-aware recommender application suggests tasks that are related to private activities such as “go to movie theater”, “go shopping”, etc.

The key statement of these scenarios is that task-lists are actually tailored to different situations of the user, even if some context conditions are the same (location in this case). In this respect, our system facilitates users to access the mobile services that fit best to their current situation, purely based on qualitative context information.

[3] <http://www.nttdocomo.co.jp/english/p_s/i/felica>
[4] <http://www.jreast.co.jp/suica/>

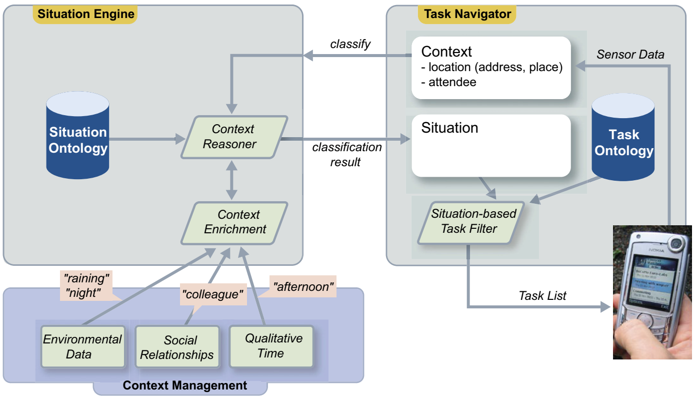

Figure 3. Architecture

4 Architecture

Figure 3 depicts the overall system architecture. The implementation contains two main parts, the situation engine and the task navigator.

The situation engine receives context information that has been collected by the task navigator on the mobile device. Furthermore, this information is enriched by context artifacts, such as environmental data, social relations between companions and a qualitative representation of time, all gathered form a distributed network of context providers. Thereupon, an axiomatized situation instance is constructed and sent to the inference engine. According to the world knowledge encoded in the situation ontology, this instance is classified and the inferred situation is propagated back to the task navigator. A subcomponent of the task navigator, the task filter, detects the most appropriate task nodes within the task ontology by matching the derived situation with the task-specific categories. Finally, a representation of the resulting task list is constructed by the task navigator and presented to the user on his mobile device for further navigation and service selections.

The task ontology stores descriptions for abstract as well as concrete tasks and their interrelations as semantic descriptions. Large and abstract tasks are thereby described by sequences of smaller subtasks. In addition, abstract tasks are annotated with enabling context conditions and concrete tasks are linked to appropriate information services via Uniform Resource Identifiers. The task structures are defined in terms of the process model of the OWL-S ontology [21]. Each task node is represented as a service class and categorized according to the high-level context concepts such as *Business_meeting*, defined within the situation ontology. The context conditions describing the applicability of a task node are thereby encoded as corresponding OWL-S service profiles. More details about our task ontology can be found elsewhere [26].

5 Context Representation and Classification

We adopted the IST MobiLife[5] Context Management Framework [7] to achieve interoperability between context sources from diverse domains by defining an XML-based context meta model. The elements of this meta model are linked to ontologies that define the basic contextual categories, used to represent qualitative aspects of context information.

[5] http:\\www.ist-mobilife.org

We refer to an ontology as a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization. Therefore, the decidability of the selected ontology language is crucial. The OWL DL fragment of the OWL fulfills this requirement, is highly expressive and has the potential to become the standard ontology language for the Semantic Web. Its selection as the ontology language of choice resulted in the construction of high-quality ontologies (i.e., ontologies that are proven consistent by fully automatic inference engines that are available for OWL DL). It is important to note that we do not propose the ontologies described hereafter as the main representation format for all aspects of context modeling, as ontologies are limited to the formulation of qualitative aspects and the available inference engines are generally weak in handing large amounts of data efficiently.

The context ontologies are composed of eight interrelated components defining more than 300 concepts, 200 properties and 300 individuals. They provide a general vocabulary for temporal and spatial concepts, agents as well as devices. Being informed by the vCard standard, the iCalendar representation and the FOAF (Friend-of-a-friend) format, an extension for the precise modeling of complex social relations has been developed. All component ontologies are integrated by a situation ontology that defines a top-level concept named *Situation* (cf. Figure 4). This concept is refined by concepts such as *Private* and *Business* by referring to concepts and relations defined in the component ontologies.

We exemplarily sketch the OWL definitions of two typical situations using standard DL syntax [1]. A person's situation is classified as *Business*, if he is either located at a business place (such as an office) or at a public place (e.g., a train station) during office hours.

$$\begin{aligned} \mathit{Business} \;:=\; & \mathit{Situation} \sqcap (\exists\, \mathit{location} \,.\, \mathit{Business\_place} \sqcup \\ & (\exists\, \mathit{location} \,.\, \mathit{Public\_place} \sqcap \exists\, \mathit{time} \,.\, \mathit{Office\_hour})) \end{aligned}$$

A person is participating a family meeting if he or she is in a private meeting situation where all participants are relatives.

$$\mathit{Family\_meeting} \;:=\; \mathit{Situation} \sqcap (\forall\, \mathit{company} \,.\, \mathit{Relative})$$

Situational reasoning is realized using a DL reasoning engine that classifies concrete individual situations w.r.t. the ontology. Let us consider the Situation 1 introduced in Section 3. First, each piece of context information such as the location (Tokyo station), the time (Sunday morning), and all companions (Dawson's boss Fiona and his project partner Gordon) are represented in terms of vocabulary formalized by the context ontologies. This requires the mapping of sensed quantitative data to qualitative representations (e.g. a timestamp is mapped to an individual in the Abox representing a Friday morning). The qualitative representations are enriched by the world-knowledge formalized in the component ontologies and are combined to an Abox individual in the situation ontology.

Computed by the reasoning engine, the direct concept type for the situation instance according to Scenario 1 is *Important_meeting*. In this case, the location of the scene is a public place (as *tokyo_station* is an instance of the concept *Station*, which in turn is a subconcept of *Public_place*) during office hours (as the individual *friday_morning* is classified as *Office_hours*) and the main actor Dawson is accompanied by his supervisor and a business partner. Similarly, the situation instance constructed for Scenario 2 is classified as *Family_meeting* as it takes place at a public location during leisure time and only relatives are detected in the proximity of Dawson.

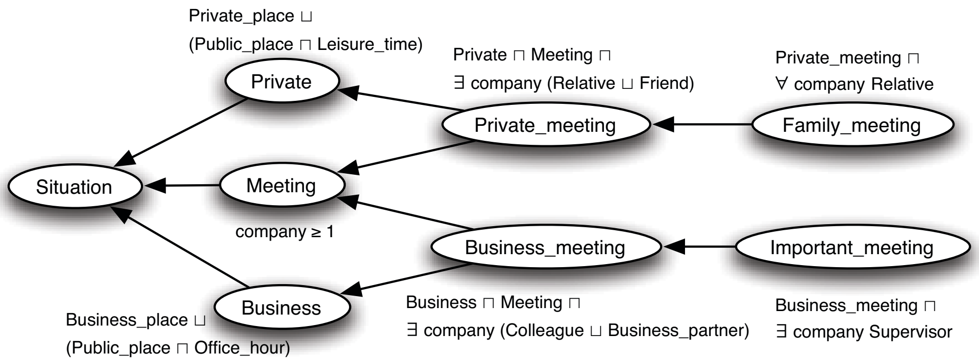

Figure 4. Situation Ontology Fragment

The situational reasoning process described above is supported by deductions in all component ontologies. For example, the agent ontology specifies in detail the semantics of social relations between people. Based on the knowledge encoded within the ontology, it can be inferred that two persons (like Dawson and Fiona) are colleagues, taking into account the transitivity of this relationship in case they have a common colleague. Similarly, even if no direct relation between Dawson and Mark is specified it can be inferred that Mark is Dawson's father in law (defined to be the father of the spouse of a person), because Dawson's wife Madeleine is known to be the child of Mark. In this case, the subproperty and inverse property specifications within the agent ontology enable this logical inference: *wife* is defined as a subproperty of *spouse* and *father* is the inverse of *child*.

6 Discussion

We integrated a situational reasoning engine into a real-world mobile service application. Our classification-based approach relies on ontology technology for the representation and reasoning on context information. As the scalable management of data is not a core feature of pure ontology-based context management and typical context models are usually rather large, we restricted its scope to high-level qualitative context elements. Lower-level context information is represented according to an XML-based meta model and managed separately. The arising reasoning problems are answered by a Description Logic (DL) [1] inference engine that provides complete reasoning support for the decidable fragment of OWL.

The use of the standard representation language OWL and the standardized reasoner interface DIG [2] (a stateless HTTP-based protocol with XML syntax) enabled us to directly compare the influence of different context ontologies and reasoners on the overall system performance. We observed that the inference technology as implemented in modern DL reasoners made significant progress during the last years. Novel optimization techniques enabled a tremendous increase in performance, and also the coverage was greatly extended. By now most systems can be accessed via DIG, and support nominals as well as Abox reasoning directly. FaCT++ and Pellet support $\mathcal{SHOIQ}(\mathcal{D})$ (OWL DL extended by qualified cardinality restrictions) and RacerPro supports $\mathcal{SHIQ}$ including approximated nominals and reasoning with concrete domains.

Nevertheless we observed several limitations in the available technology (see [18] for details). The import mechanism of OWL, which brings all triples into the importing ontology, has a limited use for the sharing and reuse of ontologies. An appropriate mechanism on the syntactic as well as the semantic level is necessary for referencing entities in another ontology without inheriting all of its complexity. Furthermore, our modeling of context ontologies would benefit from additional constructs such as qualified cardinality restrictions and a richer object property structure that would allow the specification of reflexive, irreflexive, symmetric and anti-symmetric properties as well as property chains and disjoint property axioms. Reasoning support for the DL-safe fragment [23] of SWRL [14] and for concrete domains on user defined datatypes would allow us to further enhance the quality of our situation engine. While concrete domain reasoning and support for SWRL is already available in some inference engines, and most of the requested additional language constructs are part of the OWL 1.1 draft[6] created by the ad-hoc OWL community, an improved import mechanisms as given by the $\mathcal{E}$-connection mechanism [10] and implemented in Pellet is not included.

At first, we experimented with the DIG interface to realize the communication between our application and the inference engine. However, DIG 1.1 does not support the removal of specific axioms making it necessary to re-submit the complete ontology for each request to our situation engine. This is especially awkward for our application where only a very small part of the assertional knowledge changes between two requests. As active members of the informal DIG 2.0 working group[7] we therefore propose a modular extension to the interface that supports incremental reasoning and retraction. Unfortunately, current reasoner typically only provide some kind of batch-oriented reasoning procedure. A notable exception is Racer which offers low-level retraction support for most of its statements. Still, because of the lack of algorithms for appropriately handling incremental additions as well as retractions, Racer initiates a complete reclassification after each change in the ontology. Initial empirical results, performed with an experimental version of Pellet, indicate that incremental classification algorithms for $\mathcal{SHOIN}(\mathcal{D})$ can be quite effective [28].

The ability to handle simultaneous requests is one of the key requirements in our dynamic mobile setting. However, current inference engines do not implement any transaction management. Only for Racer, support for dispatching, load balancing and caching of OWL-QL [6] queries is available via the RacerManager [8]. As OWL-QL does not support modifications of an ontology, we had to implement our own transaction management system that enables the sharing of reasoning resources between requests, but avoids the necessity to maintain a separate knowledge base for each user.

[6] ⟨http://www-db.research.bell-labs.com/user/pfps/owl⟩
[7] ⟨http://homepages.cs.manchester.ac.uk/~seanb/dig⟩

It has been observed before [17][32] that the delay caused by ontology-based inferencing easily becomes a major obstacle for realistic applications. This is especially problematic for ontologies that constantly change, because well-established optimization techniques such as tabling (used in various rule-based inference engine) cannot be applied. As a consequence of the high worst-case complexity of expressive DLs, such as $\mathcal{SHOIN}(\mathcal{D})$ underlying OWL DL, modern DL reasoners implement a suite of optimization techniques to achieve acceptable performance. The efficiency of implementations on concrete cases depends therefore on the applicability of optimizations, which varies with the language features in use. For example, the use of domain and range restrictions can lead to cycles in a Tbox for which termination of the tableaux algorithm can only be ensured by blocking. However, known blocking strategies for $\mathcal{SHOIN}$ are less effective if inverse roles are involved. On the other hand, if nominals do not occur in an ontology blocking can be realized more efficiently [13]. Therefore we avoid the use of standard ontologies, such as the $\mathcal{SHOIF}(\mathcal{D})$ entry sub-ontology of time [27]. It has to be seen how the recently suggested techniques for optimizing DL reasoning in the presence of nominals [29] perform in practice.

We optimized our initial ontologies by removing nominals and most of the domain and range restrictions. Furthermore, we reduced the number of loaded axioms and objects (especially Abox individuals) and axioms by splitting the ontology in small components and by separating ontologies in A- and Tboxes to cope with the limits of the OWL import statement. This step resulted in a performance gain of up to 1,5 seconds per request. Furthermore, we compared different retraction strategies using Racer. The simplest form of retraction is reloading of ontologies and can be accelerated by either loading from a pre-classified image or by cloning an ontology in memory. For small Aboxes cloning outperformed true retraction realized with forgot statements. However, the strategy performed best was to keep situation individuals up to a certain number (about 20 in our case) in the Abox before cloning a fresh pre-loaded Abox. Of course, keeping individuals and axioms in the Abox is only possible if they do not influence later classifications.

The time to compute our comparable simple reasoning problems is dominated by the communication overhead caused by the reasoner interface. Accessing Racer via its native API using TCP is about 1,5 times faster then the access via HTTP/DIG and even 2 times faster then the access realized with the triple-oriented framework Jena2 [3]. Naturally, we achieved the best performance by using the Pellet reasoner running in the same Java virtual machine and this way completely avoiding any external communication.

Because existing performance results of DL reasoners are often limited to static Tbox classification, we plan to perform a detailed analyze of the influence of different retraction strategies for dynamic assertional reasoning, to compare the performance of interfaces and to test the effect of the ontology size and complexity on realistic reasoning tasks. By that we hope to gain inside on how to further optimize our situation engine.

Our current prototype has only a limited support for automatic context acquisition. We plan to advance the prototype towards the use of more actual context information from the real world. Planed extensions will combine GPS-based location information with the RFID-based context tags we use currently for location tracking, as well as or short distance wireless communication technologies such as Bluetooth to detect people in proximity [19].

REFERENCES

[1] F. Baader et al., *The Description Logic Handbook*, Cambridge University Press, Cambridge, January 2003.

[2] S. Bechhofer, R. Möller, and P. Crowther, 'The DIG Description Logic Interface', in *Proc. of the Int. Workshop on Description Logics*, (2003).

[3] J. Carrol et al., 'Jena: Implementing the Semantic Web recommendations', Technical Report HPL-2004-146, HP Labs, (2004).

[4] H. Chen, T. Finin, and A. Joshi, 'The SOUPA ontology for pervasive computing', in *Ontologies for Agents*, eds., V. Tamma, S. Cranefield, and T. Finin, Springer, (July 2005).

[5] K. de Bruin and D. Fensel, 'Owl-', WSML Deliverable D20.1, (2005).

[6] R. Fikes et al., 'OWL-QL', *Journal of Web Semantics*, (2005).

[7] P. Floréen, M. Przybilski, P. Nurmi, J. Koolwaaij, A. Tarlano, M. Wagner, and M. Luther et al., 'Towards a context management framework for MobiLife', in *Proc. of the IST Summit*, (June 2005).

[8] J. Galinski et al., 'Development of a server to support the formal Semantic Web query language OWL-QL', In Horrocks et al. [15].

[9] Y. Gil, E. Motta, R. Benjamins, and M. Musen, eds. *Proc. of the 4th Int. Semantic Web Conference*, volume 3729 of *LNCS*. Springer, 2005.

[10] B. Grau, B. Parsia, and E. Sirin, 'Combining OWL ontologies using $\mathcal{E}$-connections', *Journal of Web Semantics*, **4**(1), (2005).

[11] B. Grosof, I. Horrocks, R. Volz, and S. Decker, 'Combining logic programs with Description Logic', in *Proc. of the Int. WWW Conf.*, (2003).

[12] V. Haarslev and R. Möller, 'Racer: A core inference engine for the Semantic Web Ontology Language (OWL)', in *Proc. of the 2nd Int. Workshop on Evaluation of Ontology-based Tools*, pp. 27–36, (2003).

[13] V. Haarslev, R. Möller, and M. Wessel, 'Description Logic inference technology: Lessions learned in the trenches', In Horrocks et al. [15].

[14] I. Horrocks and P Patel-Schneider, 'A proposal for an OWL rules language', in *Proc. of the Int. WWW Conf.*, pp. 723–731. ACM, (2004).

[15] I. Horrocks et al., ed. *Int. Workshop on Description Logics*, July 2005.

[16] D. Khushraj and O. Lassila, 'CALI: context awareness via logical inference', in *Proc. of the Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications*, (November 2004).

[17] J. Lee, I. Park, and S. Hyun, 'Application-oriented context pre-fetch method for improving inference performance in ontology-based context management', in *Workshop on Contexts and Ontologies*, (2005).

[18] Th. Liebig, M. Luther, O. Noppens, M. Paolucci, M. Wagner, and F.. von Henke, 'Building applications and tools for OWL', in *Proc. of the OWL Experiences and Directions WS*, (November 2005).

[19] M. Luther, S. Böhm, M. Wagner, and J. Koolwaaij, 'Enhanced presence tracking for mobile applications', In Gil et al. [9].

[20] M. Luther, B. Mrohs, M. Wagner, S. Steglich, and W. Kellerer, 'Situational reasoning – a practical OWL use case', in *Proc. of the 7th Int. Symp. on Autonomous Decentralized Systems*, (April 2005).

[21] D. Martin et al., 'OWL-S: Semantic Markup for Web Services', W3C Member Submission, The OWL Serivces Coalition, (November 2004).

[22] D. McGuinness and F. van Harmelen, 'OWL Web Ontology Language overview', W3C Recommendation, (February 2004).

[23] B. Motik, U. Sattler, and R. Struder, 'Query answering for OWL-DL with rules', in *Proc. of the Int. Semantic Web Conf.*, (2004).

[24] B. Mrohs, M. Luther, R. Vaidya, and M. Wagner et al., 'OWL-SF – a distributed semantic service framework', in *Proc. of the Workshop on Context Awareness for Proactive Systems*, pp. 67–77, (June 2005).

[25] T. Naganuma and S. Kurakake, 'Task knowledge based retrieval for service relevant to mobile user activity', In Gil et al. [9], pp. 959–973.

[26] T. Naganuma and M. Luther et al., 'Task-oriented mobile service recommendation enhanced by a situational reasoning engine', in *Proc. of the 1st Asian Semantic Web Conference (ASWC'06)*, (2006). To Appear.

[27] F. Pan and J. Hobbs, 'Time in OWL-S', in *Proceedings of the AAAI Spring Symposium on Semantic Web Services*, (2004).

[28] B. Parsia et al., 'Towards incremental reasoning through updates in OWL-DL', in *Reasoning on the Web WS*, (2006). To Appear.

[29] E. Sirin, B. C. Grau, and B. Parsia, 'From wine to water: Optimizing Description Logic reasoning for nominals', in *Int. Conf. on the Principles of Knowledge Representation and Reasoning*, (2006). To Appear.

[30] E. Sirin and B. Parsia, 'Pellet: An OWL DL reasoner', in *Proc. of the Int. Workshop on Description Logics*, pp. 212–213, (2004).

[31] D. Tsarkov and I. Horrocks, 'Ordering heuristics for Description Logic reasoning', in *Proc. of the 19th Int. Conf. on AI (IJCAI'05)*, (2005).

[32] X. Wang, Q. Zhang, T. Gu, and H. K. Pung, 'Ontology-based context modeling and reasoning using OWL', in *Proc. of the Workshop on Context Modeling and Reasoning (PerCom'04)*, pp. 18–22, (March 2004).

Integrating Multiple Contexts and Ontologies in a Pervasive Computing Framework [1]

Adrian K. Clear and Stephen Knox and Juan Ye and Lorcan Coyle and Simon Dobson and Paddy Nixon [2]

Abstract. There is a commonly accepted need for contexts and ontologies to describe the vast amounts of data that are available to pervasive computing applications. Existing contexts and ontologies are either much generalised, very application specific, or inflexible. An integrated approach is required in which new concepts can be added and related to existing ones transparently. This paper describes a novel approach to the design of a set of contexts and ontologies for context-aware pervasive computing systems. It describes a *Query Service*, that lies between applications and contextual information, which complemented by the contexts and ontologies, offers a more powerful query answering service to application developers than is currently available.

1 Introduction

Pervasive systems are interactive systems, whose behaviour must adapt to the user's changing tasks and environment using different interface modalities and devices [8]. In order to be able to adapt to its environment, the pervasive system's applications and the environmental sensors must have a common understanding of the contextual information. For this purpose, contexts and ontologies are vital. We view an ontology as an explicit modelling of the fundamental concepts of a domain that may be shared and reused. A context is an explicit model of the secondary concepts in a domain. It is more specialised than an ontology but can still be shared and reused.

To date, most ontologies for pervasive systems have been developed in a top-down manner in which the main focus is on application semantics. This leads to ad-hoc models which are neither extensible nor support interoperability [12]. On the contrary, they should be flexible in their design to support a wider range of applications and environments. Moreover, the current approach to modelling contextual data is to give it a single representation in contexts and ontologies. However, it is evident that sensors acquiring conceptually equivalent data provide different representations of such because of their nature; issues such as accuracy and heterogeneity necessitates that the data provided by these sensors are represented differently. At a common level of abstraction these representations are conceptually equivalent. The need to incorporate such relationships into the design of contexts and ontologies should be recognised and is thus the primary focus of this paper. Dealing with this issue at design time can be instrumental in the run-time inference of unknown facts from known contextual data.

Contextual data can be viewed as being part of a spectrum where data modelled by ontologies lie at one extreme, data modelled by contexts lie somewhere in the middle, and data without an explicit model lie at the other extreme. In an effort to clearly illustrate this spectrum, we propose the concept of a *semantic sphere* of pervasive system data (see figure 1). In the semantic sphere we define a set of fundamental ontologies for pervasive systems. We call this set the *core* ontology. The core ontology describes the principle concepts in a pervasive computing environment – *who*, *where* and *when*. More precisely, these are: the entities that are in the environment (people, sensors, etc.), the locations of interest, and the times of interest, respectively. All remaining data are viewed as being somewhat less general and are modelled using *application contexts* (for example, weather and music), or not modelled explicitly. Within the semantic sphere, a class definition in a context or ontology can be viewed as a *hook*. By creating an instance of one of these classes, contextual information is effectively hooked onto the context or ontology. Our contexts and ontologies are designed in such a way that semantically equivalent contextual data can be found regardless of their syntax, and coarser levels of abstraction can be inferred from finer ones. Consequently, the scope of an information search is broadened using simple relations. In this paper we also present the Query Service (QS) that has been developed so that high-level application queries can be handled transparently, and results of the appropriate level of abstraction and representation are returned to the application.

Our design approach delivers contexts and ontologies that are well-defined and flexible. Sensor developers can hook contextual data onto, or extend, an existing context or ontology. They can be easily adapted to different applications and environments. Once hooked, the contextual data is put into a distributed store, and applications can access it independently of the sensors. The novelty of this approach is the organisation of the ontologies. This, along with a powerful QS, will be very useful for the building and supporting of a large number of context-aware applications.

Our work is built within a framework called Construct, a fully-distributed and decentralised context aggregation infrastructure for pervasive computing environments [15]. Construct consists of a number of nodes that aggregate contextual data. Each node has its own data-store, and sensors register themselves with a node and inject data into it. Construct nodes gossip [6] amongst themselves to maintain a global model of the system as a whole. All information is represented using RDF as the underlying data model. Applications therefore see a soup of contextual data derived from sensors and can access it through the QS. High-level queries are passed to the QS using RDQL (RDF Data Query Language) [14]. The low level inferring is handled transparently and application-interpretable results are returned. A model of this process can be seen in figure 1.

[1] This work is partially sponsored by Science Foundation Ireland under grant numbers 05/RFP/CMS0062 and 04/RP1/I544. Adrian Clear is funded by a joint IBM/IRCSET EMBARK scholarship.

[2] The authors are with the Systems Research Group, School of Computer Science and Informatics, UCD Dublin IE (email adrian.clear@ucd.ie)

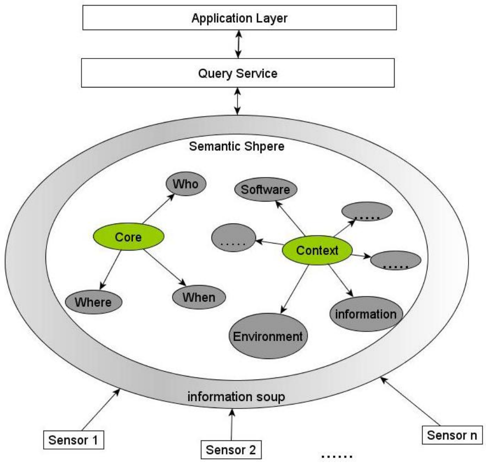

Figure 1. Contexts and Ontologies within Construct

The rest of this paper is organised as follows: Section 2 briefly illustrates the related work in the area along with the semantic web technologies that our approach depends on; in Section 3 we introduce the ontologies that are defined for the pervasive computing domain and describe some specific application contexts; Section 4 describes how the data represented in these ontologies and contexts are converted into information suitable for consumption by pervasive applications. This process is demonstrated with an example location-based application in Section 5. Finally, in Section 6 we conclude the paper and give some directions for future work.

2 Related Work

This paper addresses the issues of context modelling and context accessibility in context-aware pervasive computing systems. The area has attracted attention recently and some seminal approaches that focus on the same issues have emerged: Firstly, the work carried out by Heer et al on the liquid extension to the Context Fabric [10, 11] consists of a query service which supports distributed, continuous query processing for context data. They introduce the notion of an infospace which is a logical storage unit that may be centralised or decentralised. Once context is sensed, it is added to the appropriate infospace. Context is stored in infospaces using tuples consisting of types and values. The value can be a basic value or another infospace allowing queries to be structured as a concatenation of different types. Thus, to resolve a query involves the traversal of a string of tuples. There are drawbacks to this, however. The user is required to know the structure of the infospaces and the types of their tuples in order to make a query. Also, there is no mention of a common semantics for types that tuples may contain making interoperability difficult.

Another related concept is that of the Enactor extension to the Context Toolkit (CTK) [13, 9]. The CTK introduces Widget components which are structures that encapsulate a particular type of context acquiring sensor, for example, a location sensor. Each location sensor will have the same interface, be they an internal RF location system or GPS. This, however, allows only one level of abstraction per interface. The Enactor, which encapsulates some application logic, obviates the application developer from having to subscribe to each widget manually. It consists of a number of References which "support the declarative specification of interest in a set of CTK components through a general query package". References process queries to discoverers and automatically subscribe to any components that match. Like our approach, the low level queries are handled transparently.

Khedr et al [12] introduces context-level agreements into a multiagent pervasive computing environment. They allow user agents to specify context that is relevant to them so that the context management agent can subscribe to the appropriate context providing agents in order to have the appropriate context delivered.

All three systems support a high-level query language that decomposes requests into satisfiable responses and then returns a response to an application's request without the application needing to know the details of how the infrastructure is satisfying the response. However, there is no effort to structure the semantics of the context data to provide a more powerful query service.

Similar to the work from Chen et al on SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) [4], we use the Web Ontology Language (OWL) [3] to model our ontologies. The distinction that we make between the application contexts and the ontologies is closely based on the divide that exists in SOUPA between SOUPA Core and SOUPA Extension. Although the models that they define are quite extensive, we take the approach of organising our ontologies more effectively while keeping them simple. We also use Jena [1] which is a semantic web framework for java.

3 Contexts and Ontologies for Pervasive Computing

Numerous ad hoc ontologies have been created for pervasive computing systems to date. They have been designed with the primary goal of providing a semantics for contextual data so that a common understanding can be given to data from heterogeneous sensors along with entities in the pervasive environment. The goal of this work is to not only develop such a semantics for contextual data, but also to develop our ontologies in a way in which they can be efficiently reasoned about. The hypothesis is that different applications may require the same contextual data, but in different representations or levels of abstraction. By adding a structure to our ontologies, using relations between their contents, this reasoning over data can be done at a lower level and will thus be transparent to the application developer.

The three core ontologies of *where*, *when* and *who* are described in this section along with their general properties and relationships. These ontologies form a base model that is general enough to be used in a range of pervasive computing applications. An overview of the application contexts along with a description of the relations used in the contexts and ontologies to achieve equality and levels of abstraction are also given.

3.1 The Where Ontology

The *where* ontology describes the concept of location in a pervasive computing environment. A location may be defined as a point (*Coordinate*) or as a region (*Space*). Figure 2 shows the hierarchy of location types that are possible. Locations may be either physical, e.g. a set of GPS coordinates; or symbolic, e.g. "RivadelGarda".

Locations may be related to each other in ways that declare equivalence, e.g. `RivadelGarda=GPS (45.88,10.82)` and containment, e.g. `RoomA007 isContainedIn CS-Building`.

Section 5 gives an example of how these mappings are used to transform location data from multiple contexts into a single result at

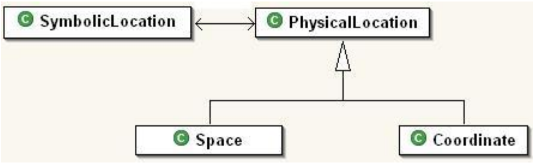

Figure 2. The *where* Ontology

the correct level of abstraction in response to a query from an application.

3.2 The When Ontology

The *when* ontology defines the concept of time in pervasive computing environment. Figure 3 illustrates this hierarchy. Time may be declared as an instant (*InstantTime*) or as a range of time (e.g. *TimeRegion*). Time may be declared as being either symbolic, e.g. `Yesterday`; or physical, e.g. `00:28, Friday 7th April 2006`. Again, there is an equivalence relationship. *TimeRegion* expresses a period of time in a tuple of `<from, to>`. We define three relationships for time: *equals*; *before*; and *after*.

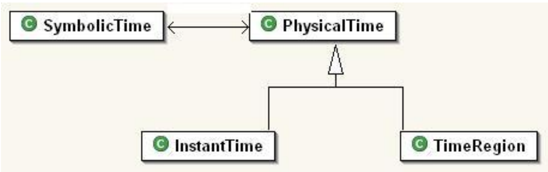

Figure 3. The *when* Ontology

3.3 The Who Ontology

The *who* ontology is different from the *when* and *where* ontologies. It describes an agent that inhabits a pervasive computing environment, e.g. a human user, intelligent agent or sensor. The *who* ontology has only one hook: an *Entity*. Every *Entity* in the system will be attached to this hook and will be uniquely identified. Each *Entity* must contain one or more *Identity* classes which are represented as attributes with values. Any piece of contextual data that identifies an agent declares itself to be representative of this token, e.g. in a tag-based location application the tag ID is mapped to an *Identity* attribute of the corresponding user *Entity*. Instances of the *who* ontology can be mapped to further, less general, contextual information such as a personal profile. Thus, by traversing the equivalence relations between *Identity* classes, any representation can gain access to contextual information regarding the agent in question.

In Section 5 we demonstrate how three representations (*Identities*) of a user from three different sensors are mapped together allowing an application to benefit from access to each of the contexts.

3.4 Application Contexts

Besides these core ontologies, there exist many other types of data that are reusable in a pervasive environment. However, they are too specific to be modelled in an ontology. We have defined application contexts for data from a number of diverse applications that we are working on. These include weather data, flight data and music data. These contexts are stored in a catalogue of data models and are available as hooks for application developers who wish to access the data of that type that are in the data store.

3.5 Transitivity and Equivalence

Contextual data can be modelled using set-theory. Referring to our core ontologies and application contexts, two relations in particular are critical to their structure; transitivity and equivalence. Consequently, there exists the notion of transitive and equivalence relations on elements of sets.

In mathematics, a binary relation *R* over a set *X* is *transitive* if it holds for all *a*, *b* and *c* in *X*, that if *a* is related to *b* and *b* is related to *c*, then *a* is related to *c*. Transitive relations strengthen the reasoning capabilities and are invaluable for certain ontology structures. For example, the *where* ontology is naturally modelled using transitive relations between different levels of abstraction of contextual data. Rooms are contained in floors which are contained in buildings, so that a result for an application query for a high level of abstraction such as "What building..." can be inferred from lower level representations of the same content.

The equivalence relation is a little simpler. An equivalence relation on a set *X* is a binary relation on *X* that is reflexive, symmetric, and transitive and it is used to group objects that are similar in some sense. Taking the *where* ontology as an example, symbolic names for locations are equivalent to their corresponding physical representations. Furthermore, in the *who* ontology, the *Identity* instances of an Entity are equivalent representations of the Entity.

To demonstrate the usefulness of these relations alone, a general query for a building name can be derived from $\langle x, y, z \rangle$ coordinates sensed by a tag-based location system by inferring the physical location that contains the coordinate (at a building level of abstraction) and finding the equivalent symbolic name.

4 Query Service

The Query Service (QS) is a layer that sits between the application layer and the data-store. It provides an interface to the application to make high-level queries on the store, and returns the results to the application in the correct level of abstraction. In order for sensor developers to take advantage of the QS functionality, they can make use of the existing ontologies and contexts so that their contextual data can be represented with the appropriate semantics and relations between levels of abstraction. The ontologies and contexts mentioned in the previous section make such a tool possible.

The QS consists of three main components; the Query Handler, the Query Executer and the Query Service Reasoner:

The Query Handler is the query interface that the QS provides to the applications of the system. Any application can use the QS by sending a high-level query to the Query Handler. When an application makes a query, the Query Handler must first determine the *known* and *unknown* facts of the query. The unknown facts are those that the application is requesting and the known facts are those that the unknown facts are requested in relation to. For example, take the query "What room is Bob in now?". In this situation, the unknown fact is the room and the known facts are Bob (the subject) and Tuesday, 11th April, 10:03am (the time that the query is made at). The next step is to find the different representations of the known and unknown facts, and query for each representation of the unknown's using the known ones as filters on the results. The Query Executer is handed all of the derived queries and the results are returned to the QS Reasoner.

The Query Executer The purpose of the Query Executer is to execute all of the low-level queries that are passed to it from the Query Handler. The Executer queries the data-store and passes the results on to the Query Service Reasoner so that it can then infer further information that the application requires. Virtual sensors may be used to derive properties that are not explicit in the contexts and ontologies. For example, a *hasLocation* property can be derived from a higher level notion of a sighting. A sighting might introduce three predicate triples to the data-store; one stating the person, one stating the time, and another stating the location.

The Query Service Reasoner The basic results returned by the Query Executer may not be of the level of abstraction required by the application. The job of the Reasoner is to reason about the results so that, if possible, they can be moulded into the representation required by the application. Currently, only bottom-up inferencing is supported as top-down inferencing would produce ambiguous or superfluous results (e.g. a building reasoning about what is contained in it could return numerous rooms). Finer levels of abstraction can be generalized to coarser ones using the relations from Section 3.5. From the query, the Reasoner knows the type, level of abstraction and representation that it must match. Using the ontologies and contexts as a reference, this match can be inferred from different representations and finer levels of abstraction. In the above example, the tag-based location system may return a coordinate which, using the *where* ontology as a reference, can infer that the coordinates are in a particular room which, in turn, is in a building as these physical spaces are defined by a set of coordinates.

Currently, a simple custom-inferencer has been implemented to reason over equivalence and transitive relations in order to seek out the required levels of abstraction. Referring to the *where* ontology, one of the transitive relations is *isContainedIn*. Equivalence relations can be defined over multiple types also. For example, a symbolic name of a location might be equal to a physical representation of a location. They are semantically equal but they are syntactically different. Consequently, a query returned for one representation can be converted to another to be of use to the application. Using these relations, the Reasoner references the appropriate ontology to locate the level of abstraction and representation that the application is looking for. If the values returned by the Query Handler are not syntactically correct the Reasoner searches for an equivalence relationship between the syntactic form that is required and the form that is returned by the Query Handler. If one exists, the Reasoner then abstracts the value to the correct level of abstraction using the transitivity relation. Once the level of abstraction is met, the representation can be mapped to the required syntactic form using the equivalence relation.

5 Application

To demonstrate the exchange of context data and ontology data in a pervasive system, we introduce a location-tracking application that queries the data-store for the location of a user. It has a semantic map defining the locations in its realm, and a list of the users of the system. It is capable of making queries for the location of a user at the level of abstraction of a room, floor or building. We use the following sensors to provide location data at different levels of abstraction.

- Ubisense [2] sensors generate coordinate location data for each tagged user with a peak level of abstraction of 30cm in 3D space.
- Bluetooth location sensors which can track location to approximately ten metres. This provides a room-level abstraction to the data-store.
- Activity sensors determine whether an individual is located at a computer by checking whether they are logged in and active at the terminal. This sensor also provides a room-level abstraction.

Each of these sensors insert data into the data-store which have the properties: *hasLocation*, *hasTime* and *hasIdentity*. The Ubisense sensor generates raw data that looks as follows: (`tagID=tag184, time=30/03/2006 13:22:13, x=13.28, y=11.82, z=0.35`). These data are hooked to the core *who*, *when*, and *where* ontologies as follows: `tagID` is hooked onto the *Identity* class of the *who* ontology; *time* is hooked onto the *InstantTime* class of the *when* ontology; and *x=13.28, y=11.82, z=0.35* are collectively hooked onto the *Coordinate* class of the *where* ontology.

When an application asks a question relating to a person's location, e.g. "What room is Bob in now?", the Query Handler takes the known and unknown terms and generates a suitable query in RDQL:

```
SELECT ?location WHERE
?person alsoKnownAs Bob
?time after (currentTime − 5)
?x hasTime ?time
?x hasIdentity ?person
?x hasLocation ?location
```

The Query Executer executes this broad query. At least three results will be found (one for each active sensor). The data that came from the Ubisense sensor might come out in the following format: `(Bob, 30/03/2006 13:22:13, (13.28, 11.82, 0.35)`. These results are passed to the Query Service Reasoner.

The required level of abstraction for the location data response is at the room level. In this example, two different levels of abstraction are returned; data at the room granularity (the data that originated at the activity sensor and Bluetooth sensor); and at the coordinate level (from the Ubisense sensor). The former results are at the correct level, and can be returned unaltered. However, the coordinate data must be raised from the coordinate level to the room level.

Figure 4 illustrates a series of steps that the Query Service Reasoner goes through to process this inconsistent data in order to return the correct level of abstraction to the application. The Query Service Reasoner starts at the level of abstraction of the available data; in this case coordinate data, and follows the transitive *isContainedIn* relation, defined in the *where* ontology, to discover whether it is contained within a defined *space*. The *equals* relation is also used to move between physical and symbolic locations. These relationships are followed until the resulting location maps to a level of abstraction that matches the original query (or it fails if there is no valid mapping — i.e. the coordinate does not match a known room). In this example, the Query Service Reasoner returns the room called "RoomA007". This is done for all available data and the results are returned to the application that made the original query.

6 Conclusions and Future Work

This paper describes a novel design approach to a set of core pervasive computing ontologies describing the concepts of *who*, *where* and *when*. These ontologies are used to ensure interoperability between data from different application contexts. Data is accessed us-

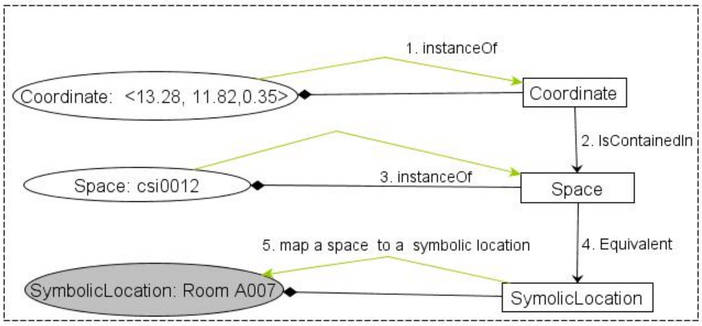

Figure 4. The process by which location data is abstracted to a higher level.

ing a specialised query service that searches for and translates appropriate data to the required level of abstraction for the query. We demonstrate this interoperability with an application that queries for location data. This data has been collected from a variety of sensors at different levels of abstraction. By using the tools in this paper the application developer does not need to be concerned with translating this data.

We describe the core ontologies. However, developers may extend from the core by implementing their own contextual models and adding them to the semantic sphere. When creating new sensors, the developer should use the preexisting contexts and ontologies but this is not required. If they enter data without an explicit model, it is available to application queries but only if they query directly against the data.

When a query is made, multiple sensors may have sensed a context which matches the query constraints. Each of these results will be returned to the application level. It is up to the application developer to process this data. One characteristic of pervasive computing environments is that sensors cannot be relied upon to always give accurate readings. Work is being done to associate a quotient of accuracy with each piece of contextual data provided by a sensor [7]. This will be available to applications and will improve the overall accuracy of an application by allowing sensor data to be fused based on the individual accuracies of the available data. We will annotate data with notions of trust [5] in the same way.

As part of our ongoing development, we intend to explore semantic translation (e.g. *adjacent* in the *where* ontology) with richer relationships in basic structures. Such semantic translation will help to support more reasoning capabilities. We intend to further develop our location-tracking algorithm to query against other types of data. We also intend to develop another application for making generic queries for pieces of data in the data-store that will assist in self-diagnosis, e.g. “tell me everything you know about *x*”, where *x* is a single piece of data.

Additionally, as a consequence of Construct’s use of gossiping to spread information between its distributed nodes, latency is a concern which must be more fully investigated. Some safeguards have to be put in place so that the occurrence of redundant data is minimised. Due to our use of backward chaining in our virtual sensor for inferencing, every query is independently dealt with by the QS. The disadvantages to this are latency and the computational cost of the same query being inferred multiple times for different applications. We will therefore be investigating the use of forward chaining, where all inferencing is done on all data when it is inserted into the data-store. In this way, the result to every satisfiable query is explicit in the data-store. This has its own problems, but would improve latency, which is paramount in pervasive systems. Truth maintenance is also a factor in pervasive systems, whereby information is inferred from lower level data. If this data is deleted or changes, it is important that this inference is still valid.

REFERENCES

[1] Jena semantic web framework. http://jena.sourceforge.net/.

[2] Ubisense. http://www.ubisense.net/.

[3] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Harrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein, ‘Owl web ontology language guide’, (2004).

[4] Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi, ‘SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications’, in *International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, MA, (August 2004).

[5] Michael Collins, Simon Dobson, and Paddy Nixon, ‘Security issues with pervasive computing frameworks’, in *Workshop on Privacy, Trust and Identity Issues for Ambient Intelligence at Pervasive 2006*, pp. 1–7, (2006).

[6] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry, ‘Epidemic algorithms for replicated database maintenance’, *SIGOPS Oper. Syst. Rev.*, **22**(1), 8–32, (January 1988).

[7] Simon Dobson, Lorcan Coyle, and Paddy Nixon, ‘Hybridising events and knowledge as a basis for building autonomic systems’, *Journal of Trusted and Autonomic Computing*, (2006). To Appear.

[8] Simon Dobson and Paddy Nixon, ‘More principled design of pervasive computing systems.’, in *EHCI/DS-VIS*, eds., Rémi Bastide, Philippe A. Palanque, and Jörg Roth, volume 3425 of *Lecture Notes in Computer Science*, pp. 292–305. Springer, (2004).

[9] A. K. Dey et al, ‘A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications’, in *Human Computer Interaction*, pp. 97–166, (2001).

[10] J. Heer, A. Newberger, C. Beckmann, and J. Hong, ‘liquid: Context-aware distributed queries’, in *Fifth International Conference on Ubiquitous Computing*, pp. 140–148, (2003).

[11] Jason I. Hong and James A. Landay, ‘An infrastructure approach to context-aware computing’, in *Human-Computer Interaction*, volume 16, (2001).

[12] M. Khedr and A. Karmouch, ‘Negotiating context information in context aware systems’, *IEEE Intelligent Systems magazine*, **19**(6), 21–29, (November/December 2004).

[13] A. Newberger and A. Dey, ‘Designer Support for Context Monitoring and Control’, (2003).

[14] Andy Seaborne. RDQL - a query language for RDF. W3C member submission, Hewlett Packard, January 2004.

[15] Graeme Stevenson, Lorcan Coyle, Steve Neely, Simon Dobson, and Paddy Nixon, ‘Construct — a decentralised context infrastructure for ubiquitous computing environments’, in *IT&T Annual Conference, Cork Institute of Technology, Ireland*, (2005).

A Context Information Manager for Pervasive Computing Environments

Jérôme Euzenat[1], Jérôme Pierson[2], Fano Ramparany[2]

Abstract. In a pervasive computing environment, heterogeneous devices need to communicate in order to provide services adapted to the situation of users. So, they need to assess this situation as their context. We have developed an extensible context model using semantic web technologies and a context information management component that enable the interaction between context information producer devices and context information consumer devices and as well as their insertion in an open environment.

1 INTRODUCTION

In a pervasive computing environment, various basic services can be provided by smart devices (e.g., sensors, actuators, human-computer interface). More advanced services can be provided when they act together and cooperate, but smarter services can only be achieved if the devices could adapt their behaviour to the user, his/her preference and his/her task, than if users have to find the specific service they want among all the smart devices.

This idea requires the perception of the environment in which devices and users interact. There are pieces of information that can be considered common to all services. In particular, spatial and temporal location as well as information related to the physical environment in which services are made available [1, 2]. These elements are part of the context in which applications operate. We are here concerned with context-aware applications, i.e., applications whose behaviour is determined to some extent by the context.

Our goal is to design a context management system general enough to be used by different pervasive computing applications, specific enough for encompassing existing services and applications, and flexible enough for supporting the dynamic addition of new devices.

First we introduce our proposal for a distributed architecture that manages context information (Section 2), then we define a context representation (Section 3) which is independent of applications and an architecture enabling their evolution. The openness of the system will require dealing with heterogeneous representations that will have to be reconciled before being used (section 4). For that purpose, we will take advantage of solutions developed for the "semantic web".

2 CONTEXTS

Context is the set of information (partly) characterizing the situation of some entity [5]. The notion of context is not universal but relative to some situation [15, 11]. This can be a physical situation (as the spatio-temporal location of some person) or functional (as the current task of the person).

Although, several scientific domains have considered the notion of context, the standpoints from which this notion is considered are different: in pervasive computing, the context of an application in terms of its physical parameters has been especially considered ; in human-computer communication, the context is most often the user task and the history of its dialogue with the computer [4]; in artificial intelligence, the context is rather considered as the conditions of validity of an assertion [14].

2.1 Context in pervasive computing

In pervasive computing, the physical context is of the utmost importance. In general, it is acquired through sensor data. These data are further elaborated into context characterization adapted to their use (for instance « high temperature » for some air conditioning controller). With regard to the sensor data (a temperature), the information has been weakened (i.e., made less precise) but is more adapted.

The various definitions of context in pervasive computing are very often related to an application or a particular domain [6, 15]. The drawback of this characterization is its reliance on the task: « high temperature » is not an absolute characterization. It depends on the use of the room (a sauna or a sleeping room). More than context, pervasive computing tends to manipulate a characterization of the context in the perspective of an application. As a consequence, it is difficult to dynamically implement non expected applications with the characterization of context made for another one.

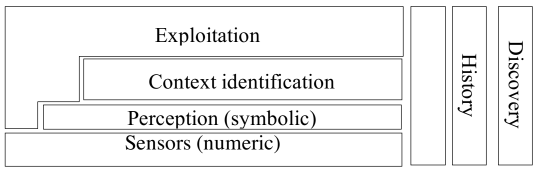

Figure 1: Model for context in pervasive computing. Data coming from sensors are aggregated and elaborated into the context used by applications (from[7]). This paper does not consider the orthogonal aspects (discovery, history and security).

However, multi-application context modelling is now understood in pervasive computing [7] and raises the issue of considering context independently from applications. Figure 1 shows the way to progressively elaborate context information from sensors to applications. We will follow this approach and this paper details the content of the perception and situation layers so that they can support the dynamic nature of the environment (new sensors and applications appear and disappear).

[1]INRIA Rhône-Alpes, France
[2]France Telecom R&D, France

2.2 Contexts in artificial intelligence

In artificial intelligence, the notion of context is, in general, concerned with the representation of information. It is used for accounting for two phenomena: the context of validity of information [16] and the efficiency of reasoning in narrower contexts [1].

John McCarthy [17] proposed a formalization of context based on context « reification » as well as the « meta-predicate » *ist*, *ist*(p,c) meaning that assertion p is true in context c. The approaches of context in artificial intelligence allow grouping knowledge in micro-theories [1] and to reason within those. In this framework (that of Cyc), the context is a more precise frame for interpreting information. This kind of approach can be used in pervasive computing in order to integrate and interpret data provided by sensors. Taking advantage of the theory associated with the sensor enables reducing the ambiguity of the data it delivers. In that view, raw data issued from sensors, are generally not weakened but rather enriched (and aggregated with other information sources allowing to further precise their interpretation). [14] describes the way to express this kind of context within the semantic web by providing each triple information on its origin (« quad »). The same model is implemented in modern RDF managers [2].

Although work from McCarthy and Guha consider contexts as independent theories related to some particular knowledge field, Fausto Giunchiglia instead considers contexts as concurrent viewpoints on the same information. He expresses the relations between contexts as « mappings » used for importing information under some context into another. This approach can be useful in pervasive computing when several information sources provide comparable information. These works found their way within semantic web tools through the C-OWL language [18]. A comparison of both approaches is made in [19].

2.3 Synthesis

In summary, pervasive computing tends to consider context as what characterizes the situation while artificial intelligence rather characterizes the information itself. More notably, Pervasive computing very often deals with the particular context of an application while artificial intelligence determines the context in function of the information source. In pervasive computing, information coming from sources is very often weakened in order to fit the application needs while artificial intelligence tends to enrich it with further information.

Of course, these approaches are rather complementary than competitors. In general, raw data can go through weakening and enrichment, thus bridging both approaches.

In pervasive computing, upgrading the environment is not an option: the environment must be designed from scratch in order to evolve. Our goal is to contribute to dealing with the dynamic evolution of context [7]. For that purpose, we design an architecture supporting the introduction of new context elements (provided from some new device) and the introduction of new applications without interruption of the environment.

This component-based context management architecture relies on a context modelling formalism based on semantic web technologies. We demonstrate how they can be used to dynamically extend the environment.

3 A CONTEXT INFORMATION MANAGEMENT COMPONENT

Pervasive Computing applications retrieve context data directly or indirectly from sensors, which are grounded in the physical environment. We propose an architecture in which applications do not need to directly connect to each sensor available and where adding a new sensor does not require all applications to be recompiled and redeployed.

3.1 Architecture

Designing an architecture for hosting context-aware services, suggests the development of a context management service for providing other services or devices with context information [6, 7, 11]. We have identified several alternative approaches for designing the target architecture. The first approach lets applications directly communicate with sensors they have an interest in. This approach requires applications to know in advance who they need to communicate with to get the information they need. Furthermore it adds complexity to the process of information aggregation, as this process should then be handled by the applications themselves and overloads sensors activity. Finally this approach makes it difficult to insert new sensors into the environment and thus doesn't comply with our flexibility requirement.

In the framework of service oriented architectures, the second approach consists of building a context management service [4] whose job is to collect sensors information and forward this information to applications that need it. This approach makes it possible to gather sensor information in a single place so that information could be easily aggregated. For example, a system that provides local temperature and atmospheric is very useful in a home environment. At a city level, the same information is useful; however it doesn't need the same degree of precision. The drawback of such a system is that it centralizes the management of context information, which is contradictory to the concept of context. More specifically, this system provides information about the activity environment (a special case of context information), however this information is not contextual as it is independent of the current task or situation, i.e. that of the client application. Moreover, with such a system, the scope of context management would be efficient in a limited area only.

We have adopted a third approach in which each device or service embeds a context management component (CMC) for maintaining context information for its own use or for the benefit of others (Figure 2). The main advantage of this approach is that new devices can join online or leave, without having to recompile or reinitialize any part of the whole environment. This component provides mechanisms for helping context-aware devices to request context information from context sensitive devices.

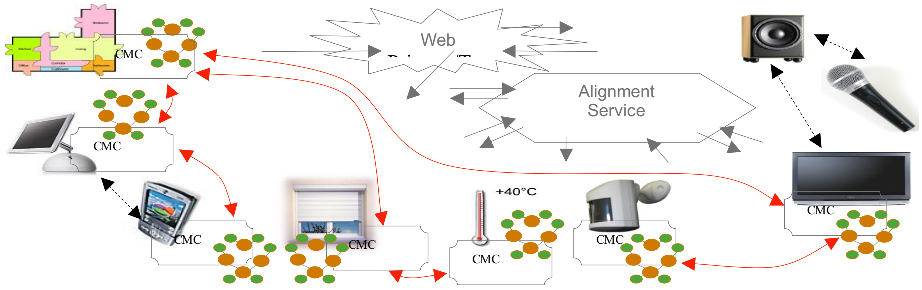

Figure 2: Each device embeds a context management component (CMC) and a semantic description of its context.

3.2 Interaction

Applications should be able to query context information they are interested in and some services should be able to provide context information, such as aggregated context information to other devices. For this purpose we design a protocol that makes the best of available services. We need to be able to identify a service, to know what kind of context information it could provide and to interact with it to get access to this information. Thus the context management component provides a few methods. In our description the first element is the query, the second is the response type:

`Id() -> URI`: The identifier of the service;
`Cl(URI) -> URI`: The class of the identified service;
`Desc(URI) -> OWL`: The description of the information that the component can provide;
`Req(RDQL) -> RDF`.

The first method allows identifying devices that are available in the environment. The identifier can then be used to contact the device. Alternatively, it could be used to get a more detailed description of the device (e.g., in case the identifier is a URI pointing to a network location where a description of the identified object is stored). A second method identifies the class (in OWL terminology) of the device. In theory, this class should be accessible from the network and once its definition is found, it provides a detailed description of the device. A third method provides the device description (or rather that of context information they provide) in an OWL formalism (OWL-S). A fourth method is used to post queries to the devices and to get the context information returned.

Thus any device is able to: find out, in its environment, services that are able to provide information relevant to its own context, get features of services that have been found (for example, measurement precision), connect to the selected service to get the information sought.

We need a language to describe the context model of heterogeneous devices so that these devices can interact in a dynamic environment.

4 OPENESS, DYNAMICS AND HETEROGENITY

The languages developed for the semantic web, and particularly RDF and OWL, are adapted to context representation in pervasive computing and particularly to the representation of dynamically evolving contexts for two reasons: these languages are open: they implement the open world assumption under which it is always possible to add more information to a context characterization; and they have been designed to work in a networked way.

4.1 Context model and language

In this dynamic pervasive computing environment, each CMC manages context information of its device. To express its context model, its needs or its capabilities, we use semantic web languages. They ensure interoperability between these heterogeneous devices.

The ground language for the semantic web is RDF (Resource Description Framework [8]). It enables expressing assertions of the form subject-predicate-object. The strength of RDF is that the names of entities (subjects, predicates or objects) are URIs (the identifiers of the web that can be seen as a generalization of URLs: http://www.w3c.org/sw). This opens the possibility for different RDF documents to refer precisely to an entity (it is reasonable to assume that a URI denotes the same thing for all of its users).

The OWL language [9], has been designed for expressing « ontologies » or conceptual models of a domain of knowledge. It constrains the interpretation of RDF graphs concerning this domain. OWL defines classes of objects and predicates and makes it possible to declare constraints applying to them (i.e., that the « output » of a « thermometer » is a « temperature »).

The context model that we use at that stage is very simple: a context is a set of RDF assertions. Interoperability is guaranteed through considering that context-aware devices are consumers and producers of RDF. However, this is not precise enough and devices may want to extract only the relevant information from context sources. For that purpose, a language like RDQL [10] is useful for querying or subscribing to context sources. In order to post the relevant queries to the adequate components, it is necessary that components publish the OWL classes of objects and properties on which they can answer.

4.2 Why ontologies?

If we can add components at any time, they may not be easily usable. Indeed, there is no a priori reason that components available, new applications and new sensors are compatible. Fortunately, knowledge representation techniques, and namely the open world assumption, makes it possible to introduce new device specifications in the environment by extending the ontology, through specifying a new concept or a property. Using ontologies to characterize the situations permits new equipment whose capabilities have not been known at the beginning to enter and new applications to benefit from these possibilities. The applications must be as general as possible describing the information they need whereas the context management system must be as precise as possible on the information it makes available. This approach enables the most specialized applications to take advantage of CMCs. The essential point is to have sufficiently generic ontologies to cover the various concepts implied in pervasive computing applications [12].

4.3 Taking advantage of heterogeneous resources

The context management system we propose makes it possible to introduce new devices in the environment by extending the ontologies in such a way that existing applications can make the best use of them. However, this view holds if all parties share the same ontology.

Unfortunately this is not always the case and agreeing on standard, universal and self contained context ontology is not a reasonable assumption. This raises the issue of matching context information with applications context information requirements. There are three alternative approaches addressing interoperability in pervasive computing environments: (*i*) A priori standardisation of ontologies, (*ii*) setting up mediators among ontologies and (*iii*) a dynamical ontology matching service. These three approaches are not incompatible and might even be jointly used. For example parties could agree on sharing common high level ontologies. Letting more specific level ontology evolve freely and independently is a strategy enabling a close account for a fast evolving domain.

As ontologies, matching services should be available for applications and context managers through network access. They provide an interface that allows the explicit handling of ontologies alignments developed in the framework of the semantic web [20]. We propose to set up one (or more) ontology matching service(s) (Figure 3). The goal of such services is to help agents (context managers in our case) to find a matching between different ontologies. These services provide mechanisms for finding out ontologies close to a given ontology, archiving (and retrieving) past alignments, dynamically computing matching between two ontologies and translating queries and responses to queries between context managers that use different ontologies [13].

5 RELATED WORKS

In pervasive computing, it is largely recognized that handling context information is essential. As we presented, there are many different management systems for context information. The one which is the nearest to what we presented here is the work on contextors [11]. It proposes a library of elements able to provide context information: it makes it possible to combine contextual information on a distributed mode. On the other hand, this system does not establish how to dynamically add devices without stopping the system or other devices. Regarding to the use of the semantic Web technologies to represent context, there are several proposals to extend the languages of the semantic Web in order to contextualize the assertions [14, 19, 2]. With regard to the use of OWL to represent the context information, [12] introduces a high level ontology of contextual information for pervasive computing.

6 CONCLUSION AND PERSPECTIVES

We specifically addressed the problem of adaptability of context management to an ever-evolving world. This is achieved by providing a distributed component-based architecture and by using semantic web technologies. Components enable the addition, at any moment, of new devices that can provide information about the context of applications. The use of RDF and OWL ensures interoperability between components developed independently by taking advantage of the open character of these technologies. Moreover, using ontology alignment modules allows dealing with the necessary heterogeneity between components. The proposed approach relies on a minimal commitment on basic technologies: RDF, OWL, and some identification protocol.

We are currently developing a demonstrator of this technology. It consists of a toolkit for developers of pervasive applications which help them deploy a distributed context management system. This toolkit provides a component for managing (searching, broadcasting and updating) context information.

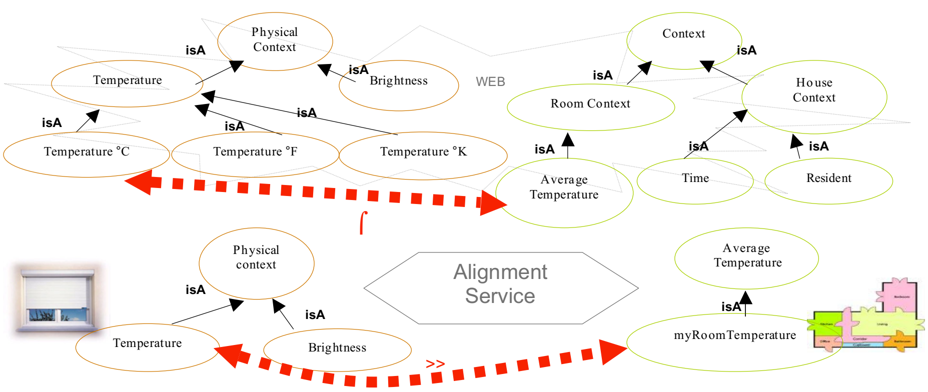

Figure 3: For finding correspondence between its model and the model of the context information provider, the window service asks to an alignment service to translate his model to another device model.

ACKNOWLEDGEMENTS

Fano Ramparany and Jérôme Pierson are partially supported by the European project Amigo (IST-2004-004182); Jérôme Euzenat is partially supported by the European network of excellence Knowledge Web (IST-2004-507482).

REFERENCES

[1] R. Guha, Contexts: a formalization and some applications, PhD thesis, Stanford university (CA US), 1991 (Technical Report STAN-CS-91-1399-Thesis et MCC ACT-CYC-423-91).

[2] O. Khriyenko, V. Terziyan, Context description framework for the semantic web, Proceedings Context 2005 Context representation and reasoning workshop, Paris (FR), 2005

[3] A. Dey, D. Salber, G. Abowd, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, *Human-Computer Interaction* 16:97-166., 2001

[4] P. Dourish, Seeking a foundation for context-aware computing, *Human-Computer Interaction*, 16(2-3), 2001.

[5] M. Chalmers, A Historical View of Context, *Computer supported cooperative work* 13(3), 223-247, 2004.

[6] A. Dey, Understanding and using context, *Personal and ubiquitous computing* 5(1):4-7, 2001.

[7] J. Coutaz, J. Crowley, S. Dobson, D. Garlan, Context is key, *Communications of the ACM* 48(3):49-53, 2005.

[8] G. Klyne, J. Carroll, Eds., Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 2004 http://www.w3.org/TR/rdf-concepts/

[9] M. Dean, G. Schreiber Eds, OWL Web Ontology Language: Reference, W3C Recommendation, 2004. *http://www.w3.org/TR/owl-ref/*

[10] A. Seaborne, RDQL — A Query Language for RDF, W3C Member submission, 2004. *http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/*

[11] J. Crowley, J. Coutaz, G. Rey, P. Reignier, Perceptual components for context aware computing, Proceedings International Conference on Ubiquitous Computing, Göteborg (SW), pp. 117-134, 2002.

[12] X. H. Wang, D. Q. Zhang, T. Gu, H. Keng Pung, Ontology based context modeling and reasoning in OWL, Proceedings 2nd International conference on pervasive computing and communication Workshop on Context Modeling and Reasoning (CoMoRea), Orlando (FL US), 2000.

[13] J. Euzenat, Alignment infrastructure for ontology mediation and other applications, Proceedings ICSOC 1st international workshop on Mediation in semantic web services, pp.81-95, Amsterdam (NL), 2005.

[14] R. Guha, R. Fikes, R. McCool, Contexts for the Semantic Web, Proceedings 3rd ISWC, Hiroshima (JP), *LNCS* 3298:32-46, 2004.

[15] H. Chen, T. Finin, A. Joshi, An Ontology for Context-Aware Pervasive Computing Environments, *Knowledge engineering review* 18(3):197-207, 2004.

[16] J. de Kleer, An assumption-based TMS, *Artificial Intelligence* 28(2):127-162, 1986.

[17] J. McCarthy, Notes on formalizing context, Proceedings 13th IJCAI, Chambéry (FR), pp. 555-560, 1993.

[18] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, H. Stuckenschmidt, C-OWL: contextualizing ontologies, Proceedings 2nd ISWC, Sanibel Island (FL US), *LNCS* 2870:164-179, 2003

[19] L. Serafini, P. Bouquet, Comparing formal theories of context in AI, *Artificial Intelligence* 155:1-67, 2004.

[20] J. Euzenat, An API for ontology alignment, Proceedings 3rd ISWC, Hiroshima (JP), *LNCS* 3298:698-712, 2004.

Semantic Interoperability in Multi-Disciplinary Domain. Applications in Petroleum Industry

Jon Atle Gulla[1] and **Darijus Strasunskas**[1] and **Stein L. Tomassen**[1]

Abstract. The petroleum industry is a technically challenging business with high investments, complex projects and operational structures. There are numerous companies and public offices involved in the exploitation of a new oil field, and there is a high degree of specialization among them. Even though standardization has been considered important in this industry for many years, there is still very little integration across phases and across disciplines. An industrially driven consortium launched the Integrated Information Platform project in 2004, in which semantic standards based on OWL and Semantic Web technologies were to be developed for the subsea petroleum industry. In this paper, we present the IIP project in more detail and discuss applications for semantic information interoperability and retrieval.

1 INTRODUCTION

The petroleum industry in Norway is technically challenging with subsea installations and difficult climatic conditions. It is industrially still quite fragmented, in the sense that there is little collaboration between phases and disciplines in large petroleum projects. There are many specialized companies involved, though their databases and applications are not necessarily well integrated with each other. Research done by the Norwegian Oil Industry Association (OLF) shows that there is a need for more collaboration and integration across phases, disciplines and companies [1]. The existing standards do not provide the necessary support for this, and the result is costly and risky projects and decisions based on wrong or outdated data.

This paper presents the Integration Information Platform (IIP) project [2] and preliminary results. The project's goal is to extend and formalize an existing terminology standard for the petroleum industry, ISO 15926. Using Semantic Web technologies, we turn this standard into a real ontology that provides a consistent unambiguous terminology for subsea petroleum production systems. However, creating and maintaining ontologies is both time-consuming and costly. Consequently, ontologies are applied for many different tasks to increase return on investment (ROI). Therefore, the IIP project focuses on reuse of ontologies in traditional vector-space information retrieval (IR) systems, in addition to rules-based notification. Considering multi-disciplinary domain and a big variation of terminology used one of the challenges is adoption of the created ontology to the document space. Finally, it is necessary to consider how ontologies will be used in those applications, i.e. application specific ontology value is an important concern in IIP.

The paper is structured as follows. In Section 2 we go through the structures and challenges in the subsea petroleum industry, explaining the status of current standards and the vision of future integrated operations. In Section 3 the IIP project is briefly introduced. Whereas in Section 4, we discuss chosen approaches. Finally, the conclusions are drawn in Section 5.

[1] Dept. of Computer & Information Science, Norwegian University of Science & Technology (NTNU), NO-7491 Trondheim, Norway;
email: Jon.Atle.Gulla@idi.ntnu.no, Darijus.Strasunskas@idi.ntnu.no, Stein.L.Tomassen@idi.ntnu.no

2 THE SUBSEA PETROLEUM INDUSTRY

The Norwegian subsea petroleum industry is a technically challenging business. Sophisticated equipment and highly competent companies are needed, and the projects tend to be both large and expensive. Many disciplines and competences need to come together in these projects, and their success is highly affected by the way people and systems are able to collaborate and coordinate their work. On the Norwegian Continental Shelf (NCS) there are traditional oil companies, specialized service companies and smaller ICT service companies. The multidiciplinarity of the industry causes in various perspectives towards the domain, and contextual usage of different terminologies. One of the challenges is to deal with contextual information and multi-perspective data integration in the multidisciplinary industry.

Both the projects and the subsequent production systems are information-intensive. When a well is put into operation, the production has to be monitored closely to detect any deviation or problems. The next generation subsea systems include numerous sensors that measure the status of the systems and send real-time production data back to certain operation centers. For these centers to be effective, they need tools that allow them to understand this data, relate it to other relevant information, and help them deal with the situation at hand. There is a challenge in dealing with all this information, but also in interpreting information that is deeply rooted in various technical terminologies.

The multitude of companies involved, with their own applications and databases, makes coordination and collaboration more important than in the past. For the industry as a whole, this severely hampers the integration of applications and organizations as well as the decision making processes in general:

- **Integration**. Even though there is some cooperation between companies in the petroleum sector, this cooperation tends to be set up on an ad-hoc basis for a particular purpose and supported by specifically designed mappings between applications and databases. There is little collaboration across disciplines and phases, as they usually have separate databases rooted in different goals, structures and terminologies. It is of course possible to map data from one database to another, but with the complexity of data and the multitude of companies and applications in the business this is not a viable approach for the industry as a whole.
- **Decision making**. A current problem is the lack of relevant high-quality information in decision making processes. Some data is available too late or not at all because of lack of integration of databases. In other cases relevant data is not found due to differences in terminology or format. And even when information is available, it is often difficult to interpret its real content and understand its limitations and premises. This is for example the case when companies report production figures to the government using slightly different terminologies and structures, making it very hard to compare figures from one company to another.

XML is already used extensively in the petroleum industry as a syntactic format for exchanging data. Over the last few years, there have been several initiatives for defining semantic standards to achieve semantic interoperability and information sharing in the business.

2.1 ISO 15926 Integration of Life-Cycle Data

ISO 15926 is a standard for integrating life-cycle data across phases (e.g. concept, design, construction, operation, decommissioning) and across disciplines (e.g. geology, reservoir, process, automation). It consists of 7 parts, of which part 1, 2 and 4 are the most relevant to this work. Whereas part 1 gives a general introduction to the principles and purpose of the standard, part 2 specifies the modeling language for defining application-specific terminologies. Part 2 comes in the form of a data model and includes 201 entities that are related in a specialization hierarchy of types and sub-types. It is intended to provide the basic types necessary for defining any kind of industrial data. Being specified in EXPRESS [3], it has a formal definition based on set theory and first order logic.

Part 4 of ISO 15926 is comprised of application or discipline-specific terminologies, and is usually referred to as the Reference Data Library (RDL). These terminologies, described as RDL classes, are instances of the data types from part 2, are related to each other in a specialization hierarchy of classes and sub-classes as well as through memberships and relationships. If part 2 defines the language for describing standardized terminologies, part 4 describes the semantics of these terminologies. There is ongoing work in the Norwegian offshore industry to provide a comprehensive standardized terminology for the petroleum industry in part 4. Part 4 today contains approximately 50.000 general concepts like motor, turbine, pump, pipes and valves.

ISO 15926 is still under development, and only Part 1 and 2 have so far become ISO standards. In addition to adding more RDL classes for new applications and disciplines in Part 4, there is also a discussion about standards for geometry and topology (Part 3), procedures for adding and maintaining reference data (Part 5 and 6), and methods for integrating distributed systems (Part 7). Neither ISO 15926 nor other standards have the scope and formality to enable proper integration of data across phases and disciplines in the petroleum industry.

2.2 The Vision of Integrated Operations

The Norwegian Oil Industry Association proposed the Integrated Operations program in 2004. The fundamental idea is to integrate processes and people onshore and offshore using new information and communication technologies. Facilities to improve onshore's abilities to support offshore operationally are considered vital in this program. Personnel onshore and offshore should have access to the same information in real-time and their work processes should be redefined to allow more collaboration and be less constrained by time and space. OLF has estimated that the implementation of integrated operations on the NCS can increase oil recovery by 3-4%, accelerate production by 5-10% and lower operational costs by 20-30% [1].

Central in this program is the semantic and uniform manipulation of heterogeneous data. Decisions often depend on real-time production data, visualization data, and background documents and policies, and the data range from highly structured database tables to unstructured textual documents. This necessitates intelligent facilities for capturing, tracking, retrieving and reasoning about data.

Figure 1 illustrates the objectives of the integrated operations initiative. Whereas we in the current situation have numerous databases that need to be mapped to each other on an ad hoc basis, we envision a semantic standard in the future that supports integration and interoperability between data from all phases and disciplines. Suppliers's applications interact with the operators' data through standardized semantic interfaces, making sure that a unified terminology is used and data is consistent and unambiguous. The implementation requirements for integrated operations include the introduction of proper standards for efficient sharing and exchange of information.

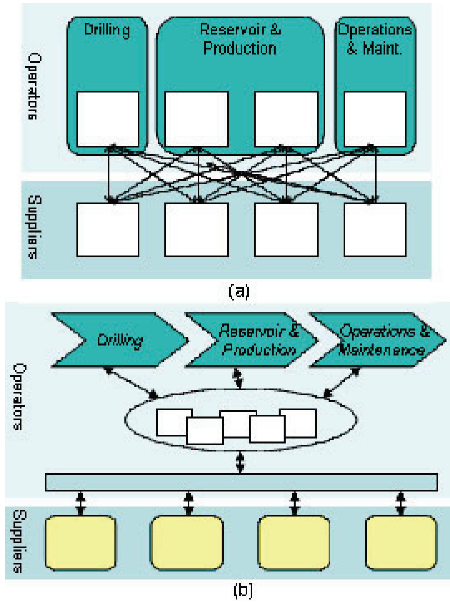

Figure 1. (a) Current situation; (b) The vision of integrated operations

3 THE INTEGRATED INFORMATION PLATFORM PROJECT

The Integrated Information Platform (IIP) project is a collaboration project between companies active on NCS and academic institutions, supported by the Norwegian Research Council (NFR). Its long-term target is to provide high quality real-time information for decision making at onshore operation centers.

The IIP project addresses the need for a common understanding of terms and structures in the subsea petroleum industry. The objective is to ease the integration of data and processes across phases and disciplines by providing a comprehensive unambiguous and well accepted terminology standard that lends itself to machine-processable interpretation and reasoning. This should reduce risks and costs in petroleum projects and indirectly lead to faster, better and cheaper decisions.

The project is identifying an optimal set of real-time data from reservoirs, wells and subsea production facilities. The OWL web ontology language is chosen as the markup language for describing these terms semantically in an ontology. The entire standard is thus rooted in the formal properties of OWL, which has a model-theoretic interpretation and to some extent support formal reasoning. A major part of the project is to convert and formalize the terms already defined in ISO 15926 Part 2 (Data Model) and Part 4 (Reference Data Library), which we will come back to in the next Section. Since the ISO standard addresses rather generic concepts, though, the ontology must also include more specialized terminologies for the oil and gas segment. Detailed terminologies for standard products and services are included from other dictionaries and initiatives (DISKOS, WITSML, ISO 13628/14224, SAS), and the project also opens for the inclusion of terms from particular processes and products at the bottom level. In sum, the ontology being built in IIP has a structure as shown in Figure 2.

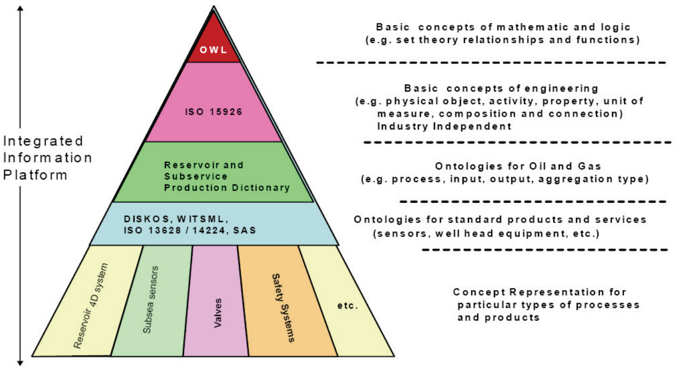

Figure 2. The standardization approach in IIP

4 APPROACH AND DISCUSSION

The success of the new ontology, and standardization work in general, depends on the users' willingness to commit to the standard and devote the necessary resources. If people do not find it worthwhile to take the effort to follow the new terminology, it will be difficult to build up the necessary support. This means that it is important to provide environments and tools that demonstrate the value of using the ontology. Intelligent ontology-driven applications must demonstrate the benefits of the new technology and convince the users that the additional sophistication pays off.

Recall, the multidisciplinary settings of the petroleum industry. The multidisciplinarity results in different views on the domain followed by vast terminology variation between disciplines, e.g. oil companies, specialized service and ICT service companies. Non-consistent usage of terminology causes the problems in documents exchange among the industrial partners (see illustration in left part of Figure 2). Furthermore, the variation in terminology may prohibit successful commitment to the ontology and its adoption in daily work routines. Therefore, we propose an approach to bridge the gap among terminologies by constructing a feature vector for each of the concepts in the ontology (see right part of Figure 3).

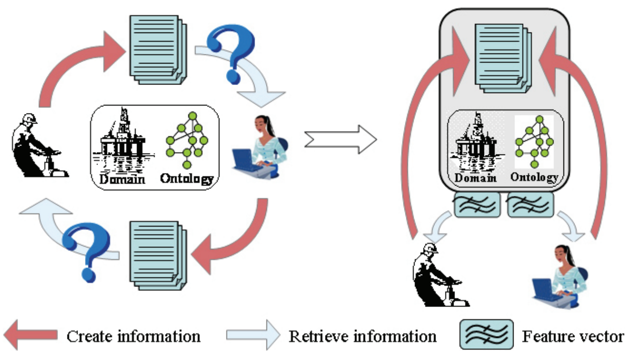

Figure 3. Multidisciplinary interpretation of domain and ontology

Development of the approach is inspired by a linguistics method for describing the meaning of objects – the semiotic triangle (known as triangle of meaning or Ogden's triangle, as well) [4]. In our approach, a feature vector connects a concept and a document collection (Figure 4), i.e., the feature vector is tailored to the terminology used in a particular collection of the documents (that is company or discipline specific). The construction of feature vector is further explained in section 4.3 and [5].

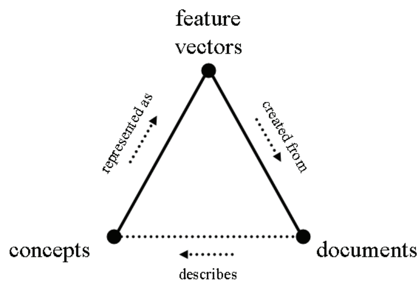

Figure 4. Explanation of a feature vector by adapted semiotic triangle

4.1 Semantic Web Technology and Interoperability

The general idea in the Semantic Web is to annotate each piece of data with machine-processable semantic descriptions. These descriptions must be specified according to a certain grammar and with reference to a standardized domain vocabulary. The domain vocabulary is referred to as an ontology and is meant to represent a common conceptualization of some domain. The grammar is a semantic markup language, as for example the OWL web ontology language recommended by W3C. With these semantic annotations in place, intelligent applications can retrieve and combine documents and services at a semantic level, they can share, understand and reason about each other's data, and they can operate more independently and adapt to a changing environment by consulting a shared ontology.

Interoperability can be defined as a state in which two application entities can accept and understand data from the other and perform a given task in a satisfactory manner without human intervention. We often distinguish between syntactic, structural and semantic interoperability [6, 7]:

- *Syntactic interoperability* denotes the ability of two or more systems to exchange and share information by marking up data in a similar fashion (e.g. using XML).
- *Structural interoperability* means that the systems share semantic schemas (data models) that enable them to exchange and structure information (e.g. using RDF).
- *Semantic interoperability* is the ability of systems to share and understand information at the level of formally defined and mutually accepted domain concepts, enabling machine-processable interpretation and reasoning.

For the Semantic Web technology to enable semantic interoperability in the petroleum industry, it needs to tackle the problem of *semantic conflicts,* also called *semantic heterogeneity*. Since the databases are developed by different companies and for different phases and/or disciplines, it is often difficult to relate information that is found in different applications. Even if they represent the same type of information, they may use formats or structures that prevent the computers from detecting the correspondence between data.

4.2 Industrial Ontologies

In recent years a number of powerful new ontologies have been constructed and applied in selected domains. This is particularly true in medicine and biology, where Semantic Web technologies and web mining have been exploited in new intelligent applications [6, 8, 9]. However, these disciplines are heavily influenced by government support and are not as commercially fragmented as the petroleum industry. Creating an industry-wide standard in a fragmented industry is a huge undertaking that should not be underestimated. In this particular case, we have been able to build on an existing standard, ISO 15926. This has ensured sufficient support from companies and public institutions. There is still an open question, though, what the coverage of such an ontology should be. There are other smaller standards out there, and many companies use their own internal terminologies for particular areas. The scope of this standard has been discussed throughout the project as the ontology grew and new companies signalled their interest. For any standard of this complexity, it is important also to decide where the ontology stops and to what extent hierarchical or complementing ontologies are to be encouraged. Techniques for handling ontology hierarchies and ontology alignment and enrichment must be considered in a broader perspective.

4.3 Ontology-driven Information Retrieval

For an Information Retrieval tool developed in IIP, we are adding a mechanism to adopt the ontology with the words used in particular discipline (i.e. by particular company) [5]. Figure 5 illustrates the overall architecture of the ontology-based information retrieval system. The individual components of the system will be given a brief account.

Feature vector miner: This component associates concept from the ontology with relevant terms from the document space. An ontology concept is a class defined in the ontology being used. These concepts are extended into *feature vectors* with a set of relevant terms extracted from the document collection using text-mining techniques. The *feature vectors* provide interpretations of concepts with respect to the document collection and needs to be updated as the document collection changes. This allows us to relate the concepts defined in the ontology to the terms actually used in the document collection.

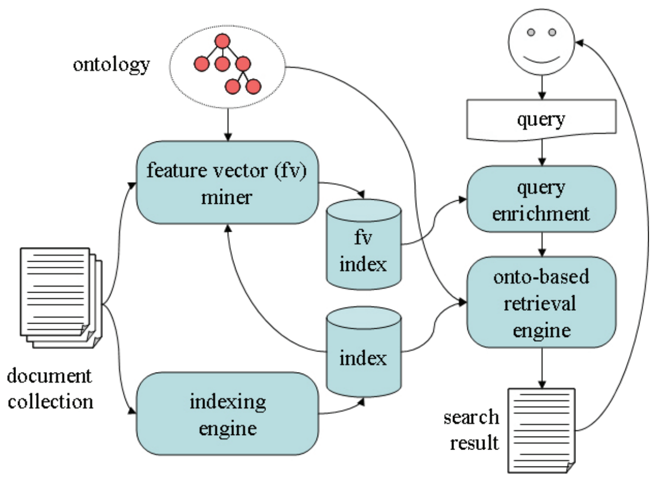

Figure 5. Architecture of ontology-driven IR system

Indexing engine: The main task of this component is to index the document collection. The indexing system is built on top of Lucene, which is a freely available and fully featured text search engine from Apache. Lucene is using the traditional vector space approach, counting term frequencies, and using *tf.idf* scores to calculate term weights in the index.

Query enrichment: This component handles the query specified by the user. The query can initially consist of concepts and/or ordinary terms (keywords). Each concept or term can be individually weighted. The concepts are replaced by corresponding feature vectors.

Onto-based retrieval engine: This component performs the search and post-processing of the retrieved results.

4.4 Rule-based Notification

Since the Semantic Web is still a rather immature technology, there are still open issues that need to be addressed in the future. One problem in the IIP project is that we need the full expressive power of OWL (OWL Full) to represent the structures of ISO 15926-2/4. Reasoning with OWL specifications is then incomplete and inference becomes undecidable [10]. Here we consider investigate the limits of inference using the ontology implemented in OWL Full. This will allow identifying possible scenarios and restrictions in using OWL Full for a such scale project. This is important, since one of the application areas is specification of rules that will be used to analyze anomalies in real-time data from subsea sensors. At that point we will need to exploit the logical properties of OWL and start experimenting with the next generation rule-based notification systems.

4.5 Application-specific Ontology Value

The quality of ontologies is a delicate topic. It is important to choose an appropriate level of granularity. In this project we have been fortunate to have an existing standard to start with. What was considered satisfactory in ISO 15926 may however not be optimal for the ontology-driven applications that will make use of the future ontology. Ultimately, we need to consider how the ontology will be used in these applications.

The ontology value quadrant [11] in Figure 6 is used to evaluate an ontology's usefulness in a particular application. The ontology's ability to capture the content of the universe of discourse at the appropriate level of granularity and precision and offer the application understandable correct information are important features that are addressed in many ontology/model quality frameworks (e.g. [12, 13, 14, 15]). But the construction of the ontology also needs to take into account dynamic aspects of the domain as well as the behavior of the application. For Ontology-driven Information Retrieval this means that we need to consider the following issues about content and dynamics [11]:

Concept familiarity. Terminologies are used to subcategorize phenomena and make semantic distinctions about reality. Ideally the concepts preferred by the user in his queries correspond to the concepts found in the ontology.

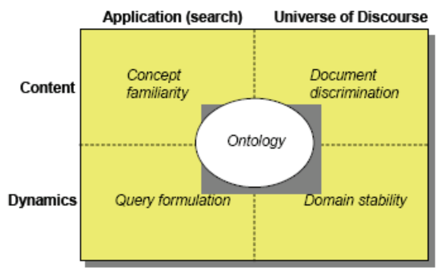

Figure 6. Ontology value quadrant [11]

Document discrimination. The structure of concepts in the ontology decides which groups of documents in the collection can theoretically be singled out and returned as result sets. Similarly, the concepts preferred by the user indicate which groups of documents she might be interested in and which distinctions between documents she considers irrelevant. If the granularity of the user's preferred concepts and the ontology concepts are perfectly compatible, combinations of these terms can single out the same result sets from the document collection.

Query formulation. The user queries are usually very short, like 2-3 words, and specialized or generalized terms tend to be added to refine a query [16]. This economy of expression seems more important to users than being allowed to specify detailed and precise user needs, as very few use advanced features to detail their query.

Domain stability. The search domain may be constantly changing, and parts of the domain may be badly described in documents compared to others. The ontology needs regular and frequent maintenance, making it difficult to depend on the availability of domain experts.

5 CONCLUSIONS

The Integrated Information Platform project is one of the first attempts at applying state-of-the-art Semantic Web technologies in an industrial setting. Existing standards are now being converted and extended into a comprehensive OWL ontology for reservoir and subsea production systems. The intention is that this ontology will later be approved as an ISO standard and form a basis for developing interoperable applications in the industry.

With the new ontology at hand, the industry will have taken the first step towards integrated operations on the Norwegian Continental Shelf. Data can then be related across phases and disciplines, helping people collaborate and reducing costs and risks. However, there are costs associated with building and maintaining such an ambitious ontology. It remains to be seen if the industry is able to take full advantage of the additional expressive power and formality of the new ontology. The work in IIP indicates that both information retrieval systems and sensor monitoring systems can benefit from having access to an underlying ontology for analyzing data and interpreting user needs.

One of the main applications developed in IIP is an ontology-driven information retrieval system [5]. Here, the concepts in the ontology are associated with contextual definitions in terms of weighted feature vectors tailoring the ontology to the content of the document collection. Further, the feature vector is used to enrich a provided query. Query enrichment by feature vectors provides means to bridge the gap between query terms and terminology used in a document set, and still employing the knowledge encoded in ontology.

Also, we can build more complete semantic descriptions of documents and add more reasoning capabilities to our information retrieval tools. We will then see if a strong semantic foundation makes it easier for us to handle and interpret the vast amount of data that are so typical to the petroleum industry.

Main future work is an inclusion of rules to be used to analyze anomalies in the real-time data from the subsea sensors. Then we will need to evaluate and investigate the logical properties of OWL and start experimenting with the next generation rule-based notification systems.

Acknowledgements

This research work is funded by the Integrated Information Platform for reservoir and subsea production systems (IIP) project, which is supported by the Norwegian Research Council (NFR). NFR project number 163457/S30.

REFERENCES

[1] OLF. Integrated Work Processes: Future work processes on the Norwegian Continental Shelf. The Norwegian Oil Industry Association. URL: http://www.olf.no/?28867.pdf (Accessed: 2006 03 05).

[2] Sandsmark, N.; Mehta, S. Integrated Information Platform for Reservoir and Subsea Production Systems. In Proceedings of the 13th Product Data Technology Europe Symposium (PDT 2004), Stockholm, (2004).

[3] International Standards Association. Industrial automation systems and integration – Product data representation and exchange. Part 11: Description methods: The EXPRESS language reference manual. URL: http://www.iso.org/iso/en/CatalogueDetailPage.CatalogoueDetail?CSNUMBER=18348. (Accessed: 2006 03 05)

[4] Ogden, C.K., and Richards, I.A. *The Meaning of Meaning*. 8th Edition, New York, Harcourt, Brace & World, Inc., (1923).

[5] Tomassen, S.L.; Gulla, J.A.; Strasunskas, D. Document Space Adapted Ontology: Application in Query Enrichment. Proceedings of 11th International Conference on Applications of Natural Language to Information Systems (NLDB'2006), Klagenfurt, Austria, Springer-Verlag, *LNCS 3999*, 46-57, (2006).

[6] Aguilar, A. Semantic Interoperability in the Context of e-Health. 2005. CDH Seminar. URL: http://m3pe.org/seminar/aguilar.pdf (Accessed: 2006 03 05).

[7] Dublin Core. Dublin Core Metadata Glossary. http://library.csun.edu/mwoodley/dublincoreglossary.html. (2004).

[8] Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. Nature Genet. **25**, 25-29, (2000).

[9] Pisanelli, D. M. (Ed.). Ontologies in Medicine. Volume 102 Studies in Health Technology and Informatics. IOS Press. (2004).

[10] Horrocks, I., Patel-Schneider, P. and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics* **1**(1), 7–26, (2003).

[11] Gulla, J. A.; Borch, H.O.; Ingvaldsen, J.E. Unsupervised Keyphrase Extraction for Search Ontologies. . Proceedings of 11th International Conference on Applications of Natural Language to Information Systems (NLDB'2006), Klagenfurt, Austria, Springer-Verlag, *LNCS 3999*, (2006).

[12] Burton-Jones, A.; Storey, V.C.;Sugumaran, V.; Ahluwalia, P. A Semiotic Metrics Suite for Assessing the Quality of Ontologies. *Data and Knowledge Engineering*, **55**(1), 84-102, (2005).

[13] Gangemi, A.; Catenacci, C.; Ciaramita, M.; Lehmann, J. Ontology evaluation and validation. An integrated formal model for the quality diagnostic task. Technical Report, ISTC-CNR, Trento, Italy, (2005). URL: http://www.loa-cnr.it/Files/OntoEval4OntoDev_Final.pdf (Accessed: 2006 04 02)

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

[15] Lozano-Tello, A.; Gomez-Perez, A. Ontometric: A method to choose appropriate ontology. *Journal of Database Management*, **15**(2), 1-18, (2004)

[16] Gulla, J.A., Auran, P.G., Risvik, K.M.: Linguistic Techniques in Large-Scale Search Engines. Fast Search & Transfer, (2002) 15 p.

A contextual personalization approach based on ontological knowledge

David Vallet[1] and **Miriam Fernandez**[2] and **Pablo Castells**[3] and **Phivos Mylonas**[4] and **Yannis Avrithis**[5]

Abstract. Combining traditional personalization techniques with novel knowledge representation paradigms, such as the ontology-based approach proposed in the Semantic Web field, is a challenging task. Personalization is a difficult problem when dealing with multimedia content and information retrieval, where context is increasingly acknowledged to be a key notion in order to make proper sense of user needs. This work focuses on contextualization within personalization in a multimedia environment. Towards that scope, we propose a novel contextual knowledge modeling scheme, and an approach for the dynamic, contextual activation of semantic user preferences to better represent user interests in coherence with ongoing user activities, e.g. in an interactive retrieval process. The application of this methodology is demonstrated using two user scenarios, and the performance results of a preliminary experiment are shown.

1 INTRODUCTION

Over the last decades, the task of personalization is related to various scientific and applied fields, with applications of techniques ranging from artificial intelligence and pattern recognition to traditional or multimedia databases and information retrieval applications [2]. One of the main issues arising is the problem of information overload, especially in the case of information retrieval that tends to select numerous multimedia documents, many of which are barely related to the user's wish [3]. This leads to other sources of information about user wishes and personalization is an approach that uses information stored in user preferences, additionally to the queries, to estimate the users' wishes and select the set of relevant documents.

In order to provide effective personalization techniques and develop intelligent personalization algorithms, it is appropriate not only to consider each user's queries/searches in an isolated manner, but also to take into account the surrounding contextual information available from prior sets of user actions. As an example, consider having some irregularities occurring in random places within a user's preferences, due to spontaneous changes of user's attention and focus. Taking into account further contextual information, the system can provide an undisturbed, clear view of the actual user's preferences, cleaned from extraordinary - according to each user's profiling information - anomalies, distractions or "noise" preferences. We refer to this surrounding information as contextual knowledge or just context.

Since several forms of context exist in the area [7], the problems to be addressed include how to represent context, how to determine it, and how to use it to influence the results of personalization. The idea behind the use of contextual information responds to the fact that not all human acts are relevant in all situations and since context is a difficult notion to grasp and capture, we restrict it herein to the notion of *ontological context*. The latter is defined as a "fuzzified" version of traditional ontologies [5]. This work is concerned with exploiting semantic, ontology-based contextual information aimed towards its use in personalization tasks. The effect and utility of the proposed invention consists of endowing a personalized retrieval system with the capability to filter and focus its knowledge about user preferences on the semantic context of ongoing user activities, so as to achieve a coherence with the thematic scope of user actions at runtime. The difficulty of successfully applying extraction of user preferences in multimedia environments, using an ontological knowledge representation constitutes this task an open and challenging issue. Finally, in the context of the Semantic Web, research efforts have resulted in the development of new knowledge representation languages, such as RDF, utilized throughout the current approach.

The rest of the paper is organized as follows: in section 2, we present the main components of the underlying knowledge infrastructure, introducing the notion of fuzzified context, as well as the use of fuzzy relations within ontologies. Section 3 deals with the problems of runtime context determination and context usage in order to influence activation of user preferences, "contextualize" them and predict or take into account the drift of preferences over time. As will be described a runtime context is represented as a set of weighted concepts from the domain ontology. How this set is determined, updated, and interpreted, will also be explained. In section 4 we provide early experimental results in the form of two user case-study examples and some conclusions are drawn in section 5.

2 ONTOLOGY-BASED KNOWLEDGE REPRESENTATION

Knowledge representation is one of the central and in some ways most fundamental notions in fields like information retrieval. Different views have been proposed and studied, and attempts have been made at determining what representation properties are important for knowledge representation in multimedia applications. However, most proposed solutions are not sufficient due to performance reasons, as well as due to the lack of accompanying contextual information. The latter forms a major limitation and it lies within the intensions of current work to manipulate and improve this kind of infor-

[1] Universidad Autonoma de Madrid, Escuela Politecnica Superior, Spain, email: david.vallet@uam.es
[2] Universidad Autonoma de Madrid, Escuela Politecnica Superior, Spain, email: miriam.fernandez@uam.es
[3] Universidad Autonoma de Madrid, Escuela Politecnica Superior, Spain, email: pablo.castells@uam.es
[4] National Technical University of Athens, Image, Video & Multimedia Laboratory, Athens, Greece, email: fmylonas@image.ntua.gr
[5] National Technical University of Athens, Image, Video & Multimedia Laboratory, Athens, Greece, email: iavr@image.ntua.gr

mation in an efficient manner. Unquestionably, design and analysis of such a task is not straight-forward and many approaches are acceptable. The term context can take many interpretations and definitions when dealing with specific application-domains [7]. This statement denotes the need for a working context interpretation applicable in personalization, since both domains will benefit from and contribute to each other. A restriction of the general notion of context is necessary, identifying the type of context suitable for user profiling and extraction of user preferences. This kind of context is defined with the aid of fuzzy algebra and ontologies, as a "fuzzified" version of traditional ontologies. We shall use the term *ontological context* from now on.

An ontology is a formal specification of a shared understanding of a domain [5]. This formal specification is usually carried out using a subclass hierarchy with relationships among the classes, where one can define complex class descriptions (e.g. in in DL [1] or OWL [8]), and use a reasoner to infer new relations among ontology elements. Given a specific domain $\mathcal{O}$ and using relations $\mathcal{R}$ and appropriate semantics, an ontology can be modeled as a set of concepts $\mathcal{C}$ together with the corresponding relations $\mathcal{R}$ between the concepts of the domain: $\mathcal{O} = \{\mathcal{C}, \{\mathcal{R}_i\}\},\ \mathcal{R}_i : \mathcal{C} \times \mathcal{C} \to \{0, 1\},\ i = 1 \dots n,\ n \in \mathbb{N}$. In this formula, $\mathcal{R}_i$ denotes the i-th relation between the concepts in the ontology.

Although in general any type of relations may be taken into consideration, in order to extract and use the desired ontological context, we define it in the means of *fuzzy ontological relations*. Fuzziness is an intrinsic property of knowledge representation, since accurate representation of real-life information is only achieved through the use of fuzzy relations. In [13] a set of basic relations is proposed that can be used to model taxonomic context hierarchies, while the relations themselves represent deeper semantics than just a taxonomic relation. Without claiming that the proposed relations are sufficient to model every type of context, we think that the relations presented in Table 1 are generic enough to form a useful basis for our personalized context model.

Table 1. Ontological relations suitable for personalization

Abbreviation	Name	Description
$Pr(x, y)$	PropertyOf	x is the property of y
$P(x, y)$	PartOf	x is part of y
$Sp(x, y)$	SpecializationOf	x is specialization of y, i.e. this corresponds to the well-know subclass relation
$Ct(x, y)$	ContextOF	x provides the context for y
$Loc(x, y)$	LocationOf	x is the location of y
$Pr(x, y)$	PropertyOf	x is the property of y

The presented relations are based on the set of semantic relations defined by the MPEG-7 standard [12]. Consequently, we may fuzzify the previous formula and describe an ontology suitable for personalization by using the following notation: $\mathcal{O}_{\mathcal{F}} = \{\mathcal{C}, \{\mathcal{R}^{\mathcal{F}}_{c_i,c_j}\}\},\ \mathcal{R}^{\mathcal{F}}_{c_i,c_j} : \mathcal{C} \times \mathcal{C} \to [0, 1],\ i, j = 1 \dots n,\ n \in \mathbb{N},\ i \neq j$. This context model forms an ontology itself, as it is compatible with the above definition. We use this "fuzzified" definition of the knowledge model in the following sections of this paper, since it is considered to be the most suitable for the modeling of information governed by uncertainty and fuzzified relations, like in the real world.

Finally, when dealing with implementation issues of the proposed context knowledge representation, we propose a specific way of representing context, following a standardized language like OWL or RDF. We smoothly integrate context's functionalities in the ontology infrastructure, i.e. we adopt enhanced characteristics available in the area of the Semantic Web, like the reification technique [10]. The proposed context model is described by pairs of concepts, represented as ontology *classes*, and relationships between the pair members, represented by *properties*. To introduce fuzziness in the approach, a degree of confidence is attached to each property. Non-existing relationships between concepts imply non-existing fuzzy relations, i.e. relations with zero confidence values are omitted. Additionally, every concept participating in the contextualized ontology has a unary degree of confidence to itself, apart from the degrees of confidence that exist between any possible class interconnections.

3 CONTEXTUAL PERSONALIZATION

Having fulfilled the first step towards contextual personalization in the form of contextual knowledge representation, the next basic step to consider is the definition of a strategy on dynamic contextualization of user preferences. Three basic principles dominate the latter:

1. representation of context as a set of domain ontology concepts that a user has "touched" or followed in some manner,
2. extension of this representation of context by using explicit semantic relations among concepts represented in the ontology
3. extension of user preferences by a similar principle

Roughly speaking, the "intersection" of the above two sets of concepts, with combined weights, are taken as the user preferences. In the following, an approximation to conditional probabilities will be utilized as an ontology-based extension mechanism. The latter is based on the existence of relations between concepts. More formally, given a finite set Ω, and $\alpha \in \Omega$, let $P(\alpha)$ be the probability that α holds some condition. We shall use this form of estimating "the probability that α holds some condition" with the purpose of extending user preferences for ontology concepts. The condition will be "the user is interested in concept α", that is, $P(\alpha)$ will be interpreted as the probability that the user is interested in concept α of the ontology. Universe Ω will correspond to a domain ontology $\mathcal{O}$ (the universe of all concepts). In the process of preferences and context expansion, a variation of constrained spreading activation (CSA) strategy is utilized [4], [11].

3.1 Semantic context for personalized content retrieval

Our model for context-based personalization can be formalized as follows: let $\mathcal{U}$ be the set of all users, let $\mathcal{C}$ be the set of all contexts, and $\mathcal{P}$ the set of all possible user preferences. Since each user will have different preferences, let $P : \mathcal{U} \to \mathcal{P}$ map each user to his/her preference. Similarly, each user is related to a different context at any given time, which we represent by a mapping $C : \mathcal{U} \times \mathbb{N} \to \mathcal{C}$, since we assume that context evolves over time. Thus we shall often refer to the elements from $\mathcal{P}$ and $\mathcal{C}$ as in the form $P(u)$ and $C(u, t)$ respectively, where $u \in \mathcal{U}$ and $t \in \mathbb{N}$. We define the contextualization of preferences as a mapping $\Phi : \mathcal{P} \times \mathcal{C} \to \mathcal{P}$ so that for all $p \in \mathcal{P}$ and $c \in \mathcal{C}$, $p| = \Phi(p, c)$.

In this context the entailment $p| = q$ means that any consequence that could be inferred from q could also be inferred from p. For instance, given a user $u \in \mathcal{U}$, if $P(u) = q$ implies that u "likes x" (whatever this means), then u would also "like x" if her/his preference was p. Now we can particularize the above definition for a specific representation of preference and context. In our model, we consider user preferences as the weighted set of domain ontology concepts for which the user has an interest, where the intensity of interest can range from 0 to 1. Given the domain ontology $\mathcal{O}$, we define the set of all preferences over $\mathcal{O}$ as $P_{\mathcal{O}} = [0,1]^{|\mathcal{O}|}$, where given $p \in \mathcal{P}_{\mathcal{O}}$, the value p_x represents the preference intensity for a concept $x \in \mathcal{O}$ in the ontology. Under the above definitions, we particularize $| =_{\mathcal{O}}$ as follows: given $p, q \in \mathcal{P}_{\mathcal{O}}, p| =_{\mathcal{O}} q \Leftrightarrow \forall x \in \mathcal{O}$, either $q_x \leq p_x$, or q_x can be deduced from p using consistent preference extension rules over $\mathcal{O}$. Additionally, we define the set of all semantic runtime contexts as $\mathcal{C}_{\mathcal{O}} = [0,1]^{|\mathcal{O}|}$. In the next sections, we propose a method to build the values of $C(u,t)$ during a user session, a model to define Φ, and the techniques to compute it. Once we define this, the activated user preferences in a given context are given by $\Phi(P(u), C(u,t))$.

3.2 Semantic extension of context

As already mentioned, the selective activation of user preferences is based on an approximation to conditional probabilities: given $x \in \mathcal{O}$ with $P_x(u) > 0$ for some $u \in \mathcal{U}$, i.e. a concept on which a user u has some interest, the probability that x is relevant for the context can be expressed in terms of the probability that x and each concept y directly related to x in the ontology belong to the same topic, and the probability that y is relevant for the context. With this formulation, the relevance of x for the context can be computed by a constrained spreading activation algorithm, starting with the initial set of context concepts defined by C.

Our strategy is based on weighting each semantic relation r in the ontology with a measure $w(r)$ that represents the probability that given the fact that $r(x,y)$, x and y belong to the same topic. We will use this as a criteria for estimating the certainty that y is relevant for the context if x is relevant for the context, i.e. $w(r)$ will be interpreted as the probability that a concept y is relevant for the current context if we know that a concept x is in the context, and $r(x,y)$ holds. Based on this measure, we use a constrained spreading activation strategy over the semantic network defined by semantic relations in the ontology, to expand the set of context concepts. As a result of this strategy, the initial context $C(t)$ is expanded to a larger context vector $EC(t)$, where of course $EC_x(t) \geq C_x(t)$ for all $x \in O$. Since $\mathcal{R}$ is the set of all relations in $\mathcal{O}$, let $\widehat{\mathcal{R}} = \mathcal{R} \bigcup \{r^{-1} | r \in \mathcal{R}\}$, and $w : \widehat{\mathcal{R}} \rightarrow [0,1]$. The extended context vector $EC(t)$ is computed by:

$$EC_y(t) = \begin{cases} C_y(t) \quad if \quad C_y(t) > 0 \\ R\left(\{EC_x(t) \cdot w(r) \cdot power(x)\}_{x \in \mathcal{O}, r \in \widehat{\mathcal{R}}, r(x,y)}\right) \end{cases}$$

where R is defined as:

$$R(X) = \sum_{S \subset \mathbb{N}_n} (-1)^{|S|+1} \prod_{i \in S} x_i$$

and $X = \{x_i\}_{i=0}^{n}$, where $x_i \in [0,1]$ and $power(x) \in [0,1]$ is a propagation power assigned to each concept x (by default, $power(x) = 1$).

3.3 Semantic preference expansion

In information retrieval two major issues need to be considered towards the efficient manipulation and exploitation of user preferences. The first thing to consider is their ability to adapt to the contextual environment, i.e. their context adaptiveness, and the second thing is the special care that needs to be taken for an overall profile consistency after application of user preferences contextualizing methodologies. Under these circumstances, a novel approach is followed: extension of preferences through ontology relations, following in general the same approach, that is used to expand the runtime context.

The idea behind this methodology is to follow the principles used for the extension of the semantic context in the previous subsection 3.2. The main difference is that here relations are assigned different weights $w'(r)$ for propagation, since the inferences one can make on user preferences, based on the semantic relations between concepts, are not necessarily the same as one would make for the contextual relevance. In general, it is expected that $w'(r) \leq w(r)$, i.e. user preferences are expected to have a shorter expansion than context has. Given an initial user preference P, the extended preference vector EP is defined by:

$$EP_y = \begin{cases} P_y, \; if \;\; P_y > 0 \\ R\left(\{EP_x \cdot w'(r) \cdot power(x)\}_{x \in \mathcal{O}, r \in \widehat{\mathcal{R}}, r(x,y)}\right) otherwise \end{cases}$$

which is equivalent to the previous formula for $EC_y(t)$, where EC, C and w have been replaced by EP, P and w', and variable t has been removed, since long-term preferences are taken to be stable along small amounts of time.

3.4 Contextual activation of preferences

After expanding of context, only preferred concepts with a context value different from zero will count for personalization. This is done by computing a contextual preference vector CP, as defined by $CP_x = EP_x \cdot C_x$ for each $x \in \mathcal{O}$, where EP is the vector of extended user preferences. Now CP_x can be interpreted as a combined measure of the likelihood that concept x is preferred and how relevant the concept is to the current context. Note that this vector is in fact dependent on user and time, i.e. $CP(u,t)$. At this point we have achieved a contextual preference mapping as defined in subsection 3.1, namely $\Phi(P(u), C(u,t)) = CP(u,t)$, where $P(u)| = \Phi(P(u), C(u,t))$, since $CP_x(u,t) > P_x(u,t)$ only when $EP_x(u)$ has been derived from $P(u)$ through the constrained spreading expansion mechanism, and $CP_x(u,t) < EP_x(u)$.

4 EXPERIMENTAL RESULTS

The contextualization techniques described in this work have been implemented within an experimental prototype. We have tested the prototype on a medium-scale corpus in order to put to trial the validity and soundness of the proposed model, tune parameters, observe the behavior of the contextualization system, and draw some empirical results. In subsection 4.1 we present an example scenario of detecting a user's preferences and in subsection 4.2 we provide some early evaluation results of our methodology.

4.1 An example scenario

An example scenario is provided in this section as an illustration of the application of the contextual personalization techniques. For the sake of clarity and space, full account of example details, such as the entire set of ontology concepts and relations involved, are omitted. Let us consider that Clio's family and friends have set up a common repository where they upload and share their pictures. Clio has not checked out the collection for quite a while, and she is willing to take a look of what images her relatives have brought from their last summer vacations. Let us also assume that the proposed framework has learned some of Clio's preferences over time, i.e. Clio's profile includes the weighted semantic preferences for domain concepts of the ontology, shown in Table 2, where *Tobby* is her brother's pet and an instance of *Dog*:

Table 2. Clio's initial preferences

$P(Clio)$	
Car	1.0
City	1.0
Sea	1.0
Tobby	1.0
Vegetation	1.0

This would define the P vector. Now suppose that Clio selects two images shown in Figure 1. As a consequence, a runtime context is built including the elements shown in Table 3, which corresponds to the C vector.

Figure 1. Clio's selection of pictures

Table 3. Runtime context built

$C(Clio, 1)$	
Construction	1.0
Flower	1.0

Now, Clio wants to see some picture of her family members, and issues the query "my family". The contextualization mechanism comes into place and the context set is expanded through semantic relations from the initial context, adding two more weighted concepts, shown in bold in Table 4.

This makes up the EC vector. Similarly, Clio's preferences are extended through semantic relations from her initial ones. The expanded preferences stored in the EP vector are the following, where we show the new concepts in bold (Table 5).

The contextual preferences are computed by multiplying the coordinates of the EC and the EP vectors one-on-one, yielding the CP vector depicted in Table 6 (concepts with weight 0 are omitted).

Comparing this to the initial preferences in Clio's profile, we can see that *Car*, *Sea* and *Tobby* are disregarded as out of context preferences, whereas *Construction* and *Flower* have been added because they are strongly semantically related both to the initial Clio's preferences and to the current context.

4.2 Evaluation of contextual personalization

In general, evaluating personalization tasks is known to be a difficult and expensive task [14], [9]. We have conducted a preliminary experimentation of the proposed contextualization techniques, in order to test the feasibility, soundness, and technical validity of the defined models. To this end, we have set up a corpus of significant size consisting of 145,316 text documents (445MB) from the CNN web site, plus the KIM publicly available domain ontology and KB [6]. The KB contains a total of 281 RDF classes, 138 properties, 35,689 instances, and 465,848 sentences. The CNN documents are annotated with KB concepts, amounting to over three million annotation links. The relation weights were first set manually on an intuitive basis, and tuned empirically afterwards by running a few trials. In order to extract precision and recall figures, we have rated the document/query/preference/context tuples manually. Needless to say, this is by no means a valid evaluation, but rather a first step to check the consistency of the models, to debug and tune the functions and parameters and to make some preliminary observations.

Since the contextualization techniques are applied in the course of a user session, a sequence of steps needs to be defined in order to put them to work. According to this, we use again a short scenario, as follows: Clio is fond of all kinds of luxurious and stylish articles. Her preferences include fancy brands such as Rolex, Maybach, Lexus, Hilton, Aston Martin, Bentley, Louis Vuitton, Sony, Apple, Rolls-Royce, Mercedes, Ferrari, Prada, and BMW, among others. Clio starts a search session with a query for news about Daimler-Chrysler and the different brands the company owns. Daimler-Chrysler owns both luxury brands as Mercedes or Maybach, and other more ordinary ones like Dodge or Setra that are not of interest to Clio. Personalization reorders the results according to Clio's preferences, showing first the documents related to Daimler-Chrysler and its brands Mercedes or Mayback, and pushing down other documents related to the lower-end brands of the company. In consequence, person-

Table 4. Clio's expanded context

$EC(Clio, 1)$	
Construction	1.0
City	**0.6**
Flower	1.0
Vegetation	**0.5**

Table 5. Clio's expanded preferences

$EP(Clio)$			
Car	1.0	***Tree***	**1.0**
City	1.0	***Road***	**0.5**
Construction	0.7	*Sea*	1.0
Dog	0.3	*Tobby*	1.0
Lake	0.8	*Vegetation*	1.0
Flower	1.0	***Water***	0.7
Plant	1.0		

Table 6. Clio's contextual preferences

$CP(Clio, 1)$	
Construction	0.7
City	0.6
Flower	1.0
Vegetation	0.5

alized search performs significantly better from the user's point of view. Since this is the first query of the session, no context exists yet and at this point there is no measure of the performance of the contextualization techniques.

In order to obtain this kind of information, Clio opens some documents in the search result, about the Mercedes brand and how Daimler-Chrysler is going to commercialize a new car model. She also opens a document about the new model Maybach 62. The context monitor extracts the concept of Mercedes from the documents opened by the user, along with the concept Maybach, since the selected documents were mainly about these two brands. Next, Clio makes a new query: "companies that trade on the New York Stock Exchange and have brands in the USA". The context is expanded to new concepts such as Daimler-Chrysler, owner of Mercedes and Maybach, along with all its brands. The query results are resorted according to the contextualized preferences of Clio. The documents that mention Daimler-Crhysler and Mercedes are pushed up in the result set. Clio still encounters other companies and brands that trade in the New York stock exchange and match her preferences, like the Sony Corporation, but these are not found semantically close to the brands in the context, and therefore get a lower sorting that other contents more in context with the previous user actions, which explains the improvement shown in Figure 2.

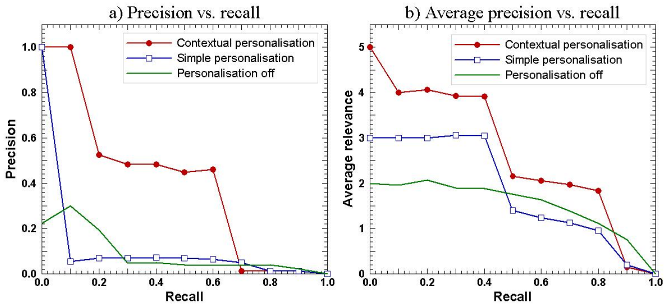

Figure 2. Comparative performance of personalized search with and without contextualization, for the query "Companies that trade on the New York Stock Exchange and commercialize a brand in the USA". The graphic a) shows the precision vs. recall curve, and b) shows the average relevance vs. recall.

5 CONCLUSIONS

Reliability is a well-known concern in the area of personalization, and one important source of inaccuracy of automatic personalization techniques is that they are typically applied out of context. I.e. although users may have stable and recurrent overall preferences, not all of their interests are relevant all the time. Instead, only a subset is usually active for them at a given time. What are the driving factors that determine this subset in a given situation is a hard question in general, all the more difficult to grasp and formalize in a computer system. Indeed, a wide range of procedural, cognitive, and environmental factors intervene in the dynamic orientation of user focus while s/he interacts with a system. The notion of context becomes elusive if one aims at a holistic approach. In this paper we propose an approximation to this problem on a specific perspective, namely based on a model of semantic runtime context of user actions, with the aim to achieve a perceivable improvement in the combination of personalization and content retrieval techniques.

As widely acknowledged, context is an increasingly common notion in information retrieval. In our approach, we combined traditional personalization techniques with novel knowledge representations, such as ontologies and reification. The use of semantic concepts, rather than plain terms, for the representation of contextual meanings, and exploitation of explicit ontology-based information attached to the concepts forms a significant novelty in the area. We also combined implicit context meanings collected at runtime, with a persistent and more general representation of user preferences. Benefit from the overall methodology is twofold: personalization techniques gain accuracy and reliability by avoiding the risk of having locally irrelevant user preferences getting in the way of a specific and focused user retrieval activity. Inversely, the pieces of meaning extracted from the context are filtered, directed, enriched, and made more coherent and meaningful by relating them to user preferences. Further insights to be drawn upon theoretic analysis, as well as the observations concerning performance and trade-offs from the experimental results of future implementation work and testing, shall provide further grounds for the analysis and evaluation of this approach.

REFERENCES

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, *The Description Logic Hand-book: Theory, Implementation and Application*, Cambridge University Press, 2002.

[2] R.A. Baeza-Yates and B.A. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press, Addison-Wesley, 1999.

[3] P.M. Chen and F.C. Kuo, *An information retrieval system based on a user profile*, Journal of Systems and Software 54, 3-8, 2000.

[4] F. Crestani, *Application of Spreading Activation Techniques in Information Retrieval*, Artificial Intelligence Review 11, 1997, pp. 453-482.

[5] T.R. Gruber, *A Translation Approach to Portable Ontology Specification*, Knowledge Acquisition 5: 199-220, 1993.

[6] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, *Semantic Annotation, Indexing, and Retrieval*, Journal of Web Sematics 2(1), 2004, pp. 49-79.

[7] Ph. Mylonas and Y. Avrithis, *Context modeling for multimedia analysis and use*, Proc. of 5th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT '05), 2005.

[8] W3C Recommendation, *OWL Web Ontology Language Reference*, http://www.w3.org/TR/owl-ref/, 10 February 2004.

[9] B. Rajagopalan and A. Deshmukh, *Evaluation of Online Personalization Systems: A Survey of Evaluation Schemes and A Knowledge-Based Approach*, Journal of Electronic Commerce Research 6(2), 2005, pp. 112-122.

[10] RDF Semantics, W3C Recommendation, *Reification, Containers, Collections and rdf:value*, http://www.w3.org/TR/rdf-mt/#ReifAndCont, 10 February 2004.

[11] G. Salton, C. Buckley, *On the Use of Spreading Activation Methods in Automatic Information Retrieval*, 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 98), Melbourne, Australia, 1988, pp. 147-160.

[12] T. Sikora, *The MPEG-7 Visual standard for content descritpion - an overview*, IEEE Trans. on Circuits and Systems for Video Technology, special issue on MPEG-7, 11(6):696-702, June 2001.

[13] M. Wallace, G. Akrivas, P. Mylonas, Y. Avrithis and S. Kollias, *Using Context and Fuzzy Relations to Interpret Multimedia Content*, Proc. of the 3rd International Workshop on Content-Based Multimedia Indexing (CBMI), IRISA, Rennes, France, September 2003.

[14] R. Wilkinson and M. Wu, *Evaluation Experiments and Experience from the Perspective of Interactive Information Retrieval*, 3rd Workshop on Empirical Evaluation of Adaptive Systems, in conjunction with the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2004), Eindhoven, The Netherlands, 2004, pp. 221-230.

Ontology Based Shape Annotation and Retrieval

Olga Symonova and **Minh-Son Dao** and **Giuliana Ucelli** and **Raffaele De Amicis** [1]

Abstract. In this paper, 3D shape retrieval methodology suited for search in special category of 3D shape is presented. The proposed approach employs a fully unsupervised segmentation algorithm to decompose 3D models into components. Shape distribution vectors describing the resulting components are extracted and together with connectivity relations identify a 3D model. The 3D-shapes we are interested in this paper are models of furniture. Ontology of furniture that we started building will be used in annotation and then key word based retrieval of furniture models. A mapping between low level features extracted by the above mentioned algorithm and ontology concepts is performed. The proposed approach bridges the gap between keyword-based approaches and query-by-example approaches by using not only the low-level features but also a domain ontology.

1 INTRODUCTION

Shape description and retrieval problem arose with the growth of available information in Internet and development of technologies allowing easy creation of 3D models. However modern search engines allow only textual search of information in Internet. This approach is not effective for graphical objects [13]. Special structures describing geometrical and/or topological characteristics were suggested to substitute verbal description of a shape. The authors of [15] group shape descriptors into three large groups: feature based methods, graph based methods and other methods which can be as well compositions of the former two approaches. We refer interested reader to [14], [11], [17], [8], [3]. In our work we use the shape descriptor proposed in [14]. The descriptor is a vector of the distribution of the function defined over the shape. As the authors of [14] examined $D2$ function of the distance between two random points of the shape gives the best results. The shape distribution based descriptor can be used for categorizing 3D models into wide classes, because it is able to detect major differences between shapes, but cannot capture detailed features.

Although geometry and topology based descriptors have improved content based 3D shape retrieval, they still deal with low-level features and this leads to a big gap between low-level and high-level features. Moreover, geometrical-based matching does not consider the semantics of the object to be retrieved [12].

The research in the field of knowledge structuring suggests to use ontology for describing knowledge of a chosen domain. [5]. The author of [7] defines ontology as a specification of a representational vocabulary for a shared domain of discourse which may include definitions of classes, relations, functions and other objects. Therefore, if we know the domain in which the 3D shapes are constructed, the ontology of the domain can be built. Then mapping between low level features and ontology concepts is performed. Finally, 3D shapes are annotated and become well-defined structure under human-perspective.

[1] GraphiTech, 38050 Villazzano (TN), Salita Dei Molini 2, Italy

2 PROPOSED METHODOLOGY

2.1 Problem of shape similarity and appropriate assumptions

Content retrieval is a difficult task which is affected by the problem of ambiguity of words and shapes. It can be explained by variety of words, images and 3D models which have equal or similar spelling, shape but different meaning in different domains. This task became even more complicated while dealing with 3D models. The file containing a 3D model often lacks any description, its name can be ambiguous, misleading or not carrying any useful information. As a result descriptors containing geometrical and/or topological information are defined to be used in shape retrieval. The research in this field has proved that searching 3D graphical objects using words has worse results than while using shape descriptors [13]. However shape descriptors do not solve the problem of shape ambiguity. According to the domain where a model is used, it can have different semantic meaning, e.g. the model with the shape of human hand can be considered as a part of human body in the domain of human models or as a glove in the domain of clothing models. To solve similar problems existing in natural language (like words with different meanings) the current research suggests to build ontologies of different domains and interpret a word within the chosen domain. In order to transfer this approach to the field of shape retrieval we build ontology for 3D models, assuming that a model can be completely described by connectivity relations between its constituents and their shape. Restricting our models' domain to the one of furniture, we explain how we build the ontology of furniture, how we extract feature vectors from shapes, and using the latter, how we annotate the model and retrieve 3D objects within the same category.

According to the chosen furniture domain we can assume that models are created using Constructive Solid Geometry (CSG) approach. Thus the furniture models are assemblies of meaningful atoms that are similar to geometric primitives. To prove that this assumption does not constrict too much the number of 3D models which can be used in the proposed approach we performed a search of 3D furniture models in Internet. We downloaded 98 furniture models from Princeton Shape Benchmark [2] and Free Stuff of 3D Cafe [1]. After analysis we found that 63% of furniture models are compound models (here we notice that 88% of models from Princeton Benchmark are compound), and 75% of compound models are models composed from geometrical primitives. We suppose that these figures can increase when a 3D database is created by designers from the same industrial domain. As consequence our assumptions will be valid for the majority of CAD models, because assembly modelling is effective approach, which allows designers to work together

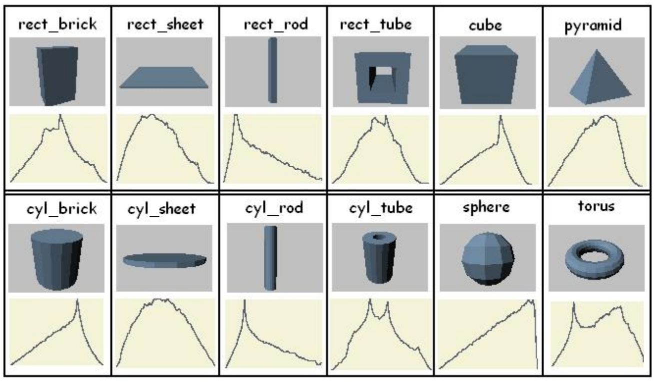

Figure 1. Geometrical primitives used for composition of 3D models and their shape distributions.

on a complex model and gives a possibility of the further reuse of designed objects.

2.2 Feature vector extraction

Given a query 3D model we start analyzing it. Considering the assumption that models are compositions of conceptual parts, we start the model analysis from its decomposition into the constituents. We load the triangle mesh, representing the given model, and then we perform its decomposition into connected components. The decomposition process has the complexity $O(|V| + |E|)$. Then we analyze the shape of each constituent of the model using the approach suggested in [14]. For each constituent we construct the vector of shape distribution. The choice of using a descriptor based on shape distribution is determined by simplicity of construction, invariance to affine transformations and good discriminative results for the models similar to geometrical primitives, like cubes, spheres, cylinders, etc [9]. According to the assumptions stated before, we consider models that are the compositions of geometrically simple objects. As a consequence we can build the finite set of the geometric primitives, which can be used to construct CAD models. For each of such geometrical primitives we extract the distribution based shape descriptor, and we label primitives with the corresponding name. At this stage the construction of the database of geometrical primitives and labeling them with corresponding names are done manually. The number of geometrical primitives which can be used for the composition of furniture model is finite, thus once the database has been constructed it can be used later without user intervention. Figure 1 illustrates which geometric primitives we have considered along with their shape descriptions. Having decomposed the given 3D model into constituents, we start to compare each part with geometrical primitives from Figure 1. The smallest distance between the vectors of shape descriptors identifies the shape of the analyzed constituent. In the current work we calculate Euclidean distance; however the other types of metrics, like Earth Mover and the Kolmogorov-Smirnov distances [14] can be used. The constituent inherits the label with the name of the most similar geometric primitive. The process continues for all parts of the model. As a result we output the vector, which has as components the names of constituent parts of the query model. For better description of the model we also analyze the connectivity relations between the parts. We compute the principal eigenvectors of the triangulations representing each part of the model and we calculate the angle between them in pairs. In this way we obtain $n \times (n-1)$ values of angles between model constituents where n is the number of connected components.

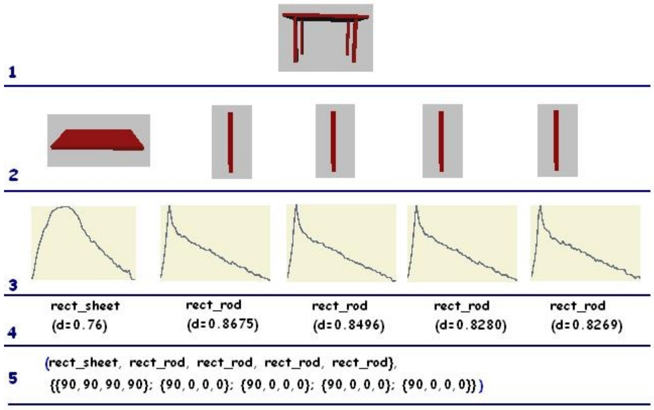

Figure 2. Feature vector extraction. 1) Input model. 2) Model decomposition. 3) Shape distributions of the model's constituents. 4) Labelling model's constituents as shape primitives. 5) Output feature vector.

To clarify the shape analysis process we consider the example of a 3D model of a table. Figure 2 shows this process. As a result we pass the feature vector identifying the query model to the Table 1, which describes the ontology of the domain . In the next chapter we explain how having the feature vector we obtain the vector of semantic labels.

2.3 Mapping feature vector to semantic labels using knowledge domain

In order to map geometrical and topological features of an object from a specific domain to semantically meaningful constituents of the object we should create a database describing all models of the domain. Table 1 illustrates the description of the models of the table of Figure 2.

Table 1. Mapping from low-level features to semantic labels.

Component	Geometrical primitive	Connectivity (pairwise angle between components	Semantic label
1	rect_sheet	{90,90,90,90}	top
2	rect_rod	{90,0,0,0}	leg
3	rect_rod	{90,0,0,0}	leg
4	rect_rod	{90,0,0,0}	leg
5	rect_rod	{90,0,0,0}	leg

The database should describe all concepts present in the ontology of the domain. Thus, querying it by the feature vector we can output as a result the vector of semantic labels. For instance taking the model of the table of the previous example, we get {*top,leg,leg,leg,leg*}, and we can pass the given semantic vector to the domain ontology in order to identify the category the model belongs to.

2.4 Ontology for shape annotation

Before building an ontology we should define its scoping, that is its domain, and its purpose, that is its intended usage [16]. In our case the domain that our ontology will formalize is that of furniture. In the first phase the intended usage of the furniture ontology is the annotation of models in the database with ontology concepts. In a second phase we want to investigate the possibility of retrieving the models by textual queries. We regard the 3D models as a syntactic domain and the ontology language as a semantic domain. An interpretation function will assign to each ”3D model” a concept from ontology. In this way we can say that a certain 3D model is a ”Chair”, while another 3D model is a ”Table”, where ”Chair” and ”Table” are concepts in our ontology. Since the classes of models are distinguished at the syntactic level by the feature vectors extracted and explained in the above sections, there are two interesting questions that an ontology based shape annotation system should answer:

1. What is the system precision? The precision of the system is defined in the well know way:

$$P = \frac{M_{Cadn}}{M_{adn}} \times 100\% \tag{1}$$

   where M_{Cadn}- is the number of correctly annotated models and M_{adn} is the number of annotated models. Since the system is still not fully operational we cannot quantify its precision, but we can make an interesting observation. The upper boundary of what can be achieved is already known. If the properties that distinguish two ontology concepts cannot be mapped to distinct sets of syntactic features that the above component can extract then the system will fail to correctly annotate the models. Let's suppose for example that there are in our ontology two concepts named ”YellowChair” and ”BlueChair”. Both concepts have as their superclass the concept ”chair” and they are distinguished only by the color they have: respectively yellow and blue. Because the above mentioned algorithm cannot extract the color of an object the system will fail to correctly annotate ”BlueChair” and ”YellowChair” models. However, assuming that for designers the shape of a model is a more important matter than its color, we suppose that the feature vector extracted on the previous step completely describes a model.
2. The second relevant parameter is the recall of the system.

$$R = \frac{M_{adn}}{M_T} \times 100\% \tag{2}$$

   where M_T is the total number of models we have. If all models are well formed the recall will be 100%.

We started building the furniture ontology using Wordnet Domains [4]. Developed at IRST, Wordnet Domains, is PWN (Princeton Wordnet) 1.6 [6] augmented with a set of Domain Labels. PWN 1.6 synsets have been semi-automatically linked with a set of 200 domain labels taken from Dewey Decimal classification, the world most widely used library classification system. The domain labels are hierarchically organized and each synset received one or more domain labels. We are interested in the synsets that are annotated with the domain ”furniture”. Because PWN is a linguistic resource and many concepts found there are not suitable for building an ontology of furniture we want to make use in our work of other ontologies and specialized thesauri.

We decided to encode our ontology in OWL language. At the moment the ontology is a simple taxonomy enriched with a relation ”hasPart” that specifies the parts of objects in the furniture domain. We make use also of cardinality restrictions as the following example, which describes the entry for the concepts ”BackRestChair” and Back Rest Chair ”BackRestChairWithFourLegs”, shows:

```
<owl:Class rdf:ID="BackRestChair">
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:onProperty>
         <owl:ObjectProperty rdf:
                         ID="hasPartLeg"/>
       </owl:onProperty>
       <owl:someValuesFrom rdf:
                         resource="#Leg"/>
     </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:someValuesFrom rdf:
                    resource="#BackRest"/>
       <owl:onProperty>
         <owl:ObjectProperty rdf:
                    ID="hasPartBackRest"/>
       </owl:onProperty>
     </owl:Restriction>
   </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:
            ID="BackRestChairWithFourLegs">
   <rdfs:subClassOf rdf:
                resource="#BackRestChair"/>
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:onProperty>
         <owl:ObjectProperty rdf:
                      about="#hasPartLeg"/>
       </owl:onProperty>
       <owl:cardinality rdf:datatype=
       "http://www.w3.org/2001/XMLSchema#int"
       >4</owl:cardinality>
     </owl:Restriction>
   </rdfs:subClassOf>
 </owl:Class>
```

The above OWL representation says that a ”BackRestChair” has as a part exactly one ”BackRest” and that a ”BackRestChairWithFourLegs” IS-A ”BackRestChair” and has exactly four legs. The only kind of inference needed in the example is the ”inheritance” of properties from super-classes to their subclasses.

2.5 Retrieval through annotations

After we annotated the 3D models with ontology concepts users have two possibilities. First they can make retrieval of 3D objects by textual query. A query can be typed by the user or can be formed by ontology browsing. For example a user interested in barber chair models can input the concept in a text box. Alternatively he can browse the ontology and select the appropriate concept. The system will answer the user query by returning all the models annotated with the input concept or with a subconcept of the input concept. An enhanced retrieval system based on textual queries can take advantage of Boolean operators.

The second possibility is to query by an example model. Here a user can browse all models within the category of the input model and autonomously search for more similar models. Such approach groups all objects into quite large classes. The other way to search for similar models is to find the smallest dissimilarity measure (i.e. the smallest distance value) between feature vectors of the constituents of a sample model and corresponding parts of models from the same category. Such approach reduces the number of comparisons needed to retrieve similar models. Furthermore, as was pointed out in [10] the descriptors based on shape distribution do not give good discriminative results for models with detailed shape properties. Decomposition of the model and shape understanding allows to perform comparison between each constituent part separately. As a result the overall dissimilarity measure will be the sum of dissimilarities between corresponding constituent parts.

3 CONCLUSIONS AND FUTURE WORK

In the current work we presented the methodology for the the new synthesis of shape description and ontology-based annotation and retrieval. Performing shape analysis we decompose a 3D model into its constituent and we analyze the shape and connectivity between each of the parts of the model. As a result we output the feature vector describing the 3D model. Using a database defining all concepts of the ontology of the given domain (here furniture), we map the extracted feature vector to the vector of semantic labels. Finally, the ontology of the considered domain will be used in model annotation and then key word based retrieval of furniture models. The proposed method offers two options to the user: textual query and query by a sample model. As a result, the proposed method succeeded in term of shape-to-text (shape annotation) and text-to-shape (query shape by text) schemes. In the future, the database will be enriched not only in terms of number of 3D models but also by number of other specific domains. Beside that, the ontology will be constructed in more details to improve the accuracy of the query process.

ACKNOWLEDGEMENTS

We thank Eduard Barbu for the useful discussions and comments on ontology construction.

This work has been supported by the CFP6 IST NoE 506766 AIM@SHAPE.

REFERENCES

[1] 3d cafe. `http://www.3dcafe.com`.

[2] Princeton shape benchmark. `http://shape.cs.princeton.edu/search.html`.

[3] Marco Attene, Silvia Biasotti, and Michela Spagnuolo, 'Shape understanding by contour driven retiling', *The Visual Computer*, **19**(2-3), 127–138, (2003).

[4] Magnini Bernardo and Cavaglia Gabriela, 'Integrating subject field codes into wordnet', in *2nd International Conference on Language Resources & Evaluation*, Athens, Greece, (May-June 2000).

[5] Balakrishnan Chandrasekaran, John R. Josephson, and Richard V. Benjamins, 'What are ontologies, and why do we need them?', *IEEE Intelligent Systems*, **14**(1), 20–26, (1999).

[6] Christiane Fellbaum (Ed.). Wordnet: An electronical lexical database. MIT Press, May 1998.

[7] Thomas R. Gruber, 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, **5**(2), 199–220, (1993).

[8] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii, 'Topology matching for fully automatic similarity estimation of 3d shapes', in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, (August 2001).

[9] Taesik Hong, Kunwoo Lee, Sungcnan Kim, Chongnam Chu, and Hyunchan Lee, 'Similarity comparison of mechanical parts', *Computer-Aided Design and Applications*, **2**(6), 759–769, (2005).

[10] Cheuk Yiu Ip, Daniel Lapadat, Leonard Sieger, and William C. Regli, 'Using shape distributions to compare solid models', in *SMA '02: Proceedings of the seventh ACM symposium on Solid modeling and applications*, New York, NY, USA, (2002).

[11] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz, 'Rotation invariant spherical harmonic representation of 3d shape descriptors', in *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Aire-la-Ville, Switzerland, Switzerland, (June 2003).

[12] Patrick Min, *A 3D Model Search Engine*, Ph.D. dissertation, Princeton University, January 2004.

[13] Patrick Min, Michael Kazhdan, and Thomas A. Funkhouser, 'A comparison of text and shape matching for retrieval of online 3d models', in *Research and Advanced Technology for Digital Libraries: 8th European Conference*, Bath, UK, (September 2004).

[14] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin, 'Matching 3d models with shape distributions', in *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, Washington, DC, USA, (2001).

[15] Johan W.H. Tangelder and Remco C. Veltkamp, 'A survey of content based 3d shape retrieval methods', in *Shape Modeling International*, Genoa, Italy, (June 2004).

[16] Mike Uschold and Martin King, 'Towards a methodology for building ontologies', in *IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, (August 1995).

[17] Dejan V. Vranic and Dietmar Saupe, '3d shape descriptor based on 3d fourier transform', in *ECMCS '01: Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services*, Budapest, Hungary, (September 2001).

Mapping contexts to vocabularies to represent intentions

Rallou Thomopoulos[1] and **Marie-Laure Mugnier**[2] and **Michel Leclère**[3]

Abstract.
In the framework of multi-target use of a given ontology, this paper proposes a representation of vocabularies based on the identification of elementary vocabularies, which can be equivalently defined using specializations of the "kind of" relation. It defines a way of combining contexts and vocabularies that allows context-specific querying.

1 INTRODUCTION

A given assertion holds in a given "context". This single affirmation can be interpreted in various ways, leading to a disparate literature about contexts. We can note two main considerations: (i) a given assertion can lead to several interpretations due to different meanings of terms, depending on the context [1, 4, 8]; (ii) the same interpretation can have different truth values in different contexts [9, 6, 12].

In this paper, our concern is to represent that, for the same piece of information, different descriptions will be given, different aspects will be highlighted, depending on the context, which can be seen as the target the piece of information will be used for (for which public and/or in which purpose). That is to say, different assertions will be used to describe the same piece of information, not because of the ambiguity of terms, nor due to the relativity of truth, but because different aspects will be important to retain, depending on the intention of the message vehiculated in each context. As a consequence, the vocabulary used in each context should be appropriate. Not all terms of the domain ontology are in accordance with the purposes of a given context: the presence of unappropriate terms, that do not conform to the intended use of information, can reveal a possible diversion out of the scope of the context, and thus not be pertinent, not understandable or not useful. For example, information intended for general public should not be too technical, terms that translate a judgement (positive, bad, ...) are expected in evaluation contexts, etc.

The aim of this paper is to propose a way of representing vocabularies and associating them with contexts. The examples, although simplified, come from a real-world application in food science. The paper is built as follows. Section 2 presents related work on contexts and ontologies. Section 3 defines the proposed representation of vocabularies. Section 4 proposes a mapping between contexts and vocabularies and shows context-specific querying that ensues.

2 RELATED WORK

2.1 Context representation

The context model we use is based on the definition of contexts as nesting types [2] in the conceptual graph model, which is a knowledge representation model based on labelled graphs. The conceptual graph model is composed of two parts: the *support*, which contains the terminological knowledge – and constitutes a part of the domain ontology –, and the *conceptual graphs*, which contain the assertional knowledge. Figure 1 shows a part of the set of concept types, noted T_C, which is part of the support.

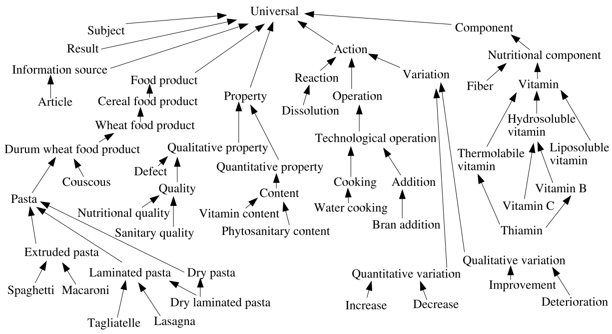

Figure 1. Part of the "food science" concept type set

A way of representing contexts in this model by structuring knowledge into levels has been descriptively introduced by [11] and furtherly studied e.g. in [3, 10]. The formalization of [2] defines a logically founded knowledge representation formalism based on nested graphs, thus providing operations for reasoning with nested graphs.

At first level, a conceptual graph gives an overall description of a fact. Zooming in on certain concept vertices provides more details, also described by conceptual graphs. A conceptual graph that is nested in a concept vertex is thus described in the context defined by this concept. Typed nestings [2] allow specifying the relationship (description, explanation, ...) between the surrounding vertex and one of its descriptions. A new type set is thus added to the support, the *set of nesting types*. In the following, a context is considered to be represented as a nesting type and expresses the target (public and/or purpose) the nested piece of information is intended for.

An example of nested conceptual graphs, built using the concept type set of Figure 1, is given in Figure 2. It represents the following piece of information: "an article, whose subject is a wheat food product that is cooked in water, has a result, whose nutritional observation is that the vitamin content of this wheat food product decreases, whose biochemical explanation is that this wheat food product contains hydrosoluble vitamin that is dissolved, and whose nutritional evaluation is that the nutritional quality of this wheat food product is deteriorated".

The set of conceptual graphs is partially pre-ordered by the *specialization relation* (noted $\leq$), which can be computed by the *pro-*

[1] INRA (IATE Joint Research Unit) / Associate Researcher of LIRMM, Montpellier, France, email: rallou@ensam.inra.fr
[2] LIRMM, Montpellier, France, email: mugnier@lirmm.fr
[3] LIRMM, Montpellier, France, email: leclere@lirmm.fr

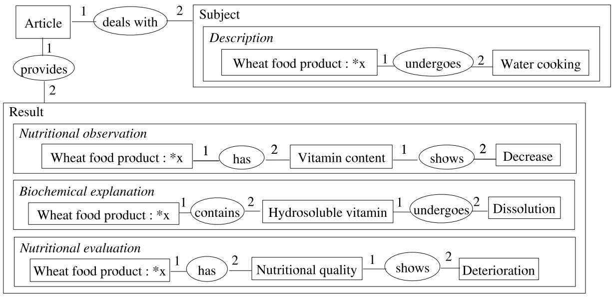

Figure 2. An example of nested conceptual graphs

jection operation (a graph morphism allowing the restriction of the vertex labels). The projection is a ground operation in the conceptual graph model since it allows the search for answers, which can be viewed as specializations of a query (see Section 4.2).

2.2 Ontology structure

The question of combining different vocabularies is a major concern of ontology integration. Several studies (e.g. [7, 5]) have proposed distinguishing between different kinds of terminologies according to their level of generality, the top-level being usable for large communities of users, whereas the more specific ones are obtained by specializing the more general levels and used for more specific needs.

However, pertinent vocabulary, for a given use, does not always depend on its depth in the ontology. An example is the following. To express information intended for a general public, we can note that, besides top-level concept types (see Figure 1), several other concept types are pertinent because they correspond to commonly used categories (Spaghetti, Lasagna ...), although they are more specific than concept types that correpond to technical categories (Extruded pasta, Laminated pasta ...) and hence cannot be used. In this example, this is due to the fact that Spaghetti or Lasagna are appellations, they do not explicitely express technical criteria.

3 VOCABULARY REPRESENTATION

Due to this consideration, an alternative basis to characterize pertinent vocabulary for a given use, other than its depth in the ontology, seems coherent to us. We propose a construction of vocabularies based on the specialization criteria used to obtain the concept types that compose them (appellation, technology, ...). We will firstly define “vocabularies”, then propose two equivalent ways of constructing them.

3.1 Identification of elementary vocabularies

Definition 1 *A vocabulary is a subset of the concept type set T_C. A vocabulary V_1 is more specific than V_2 if $V_1 \subseteq V_2$.*

According to this definition, a vocabulary composed of top-level concept types is not more general than a vocabulary composed of more specific concept types: the two vocabularies are not comparable. The most general vocabulary is T_C, as it contains all the others. This is in accordance with conceptual graph specialization and projection (illustrated in Section 4.2).

As mentioned in previous works (see part 2.2), in practice ontologies are constructed by successive specializations from top to bottom level. Moreover considering that several direct specializations of a given concept type can have related meanings seems sensible. To conserve these notions, we consider that vocabularies are composed of elementary vocabularies built by successive specializations, in a top-down way, of the concept type set.

Definition 2 *T_C is partitioned into a set of elementary vocabularies V_i built as follows:*
- V_0 is composed of the Universal *concept type;*
- For $n > 0$, V_n is obtained by defining specializations of concept types of one elementary vocabulary V_k ($k < n$), or common specializations of several given elementary vocabularies, through a given specialization criterion[4] (noted crt).

An example is given in Figure 3 for a small part of the set of concept types. The criterion used for each vocabulary is noted in brackets. In this example, each elementary vocabulary is built by specializing one preceding elementary vocabulary.

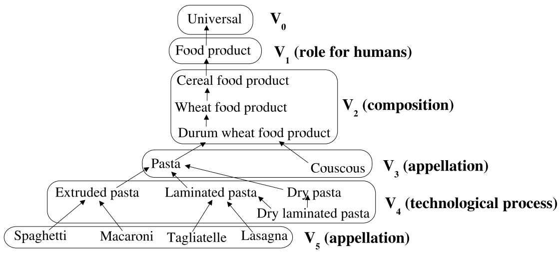

Figure 3. Example of vocabulary construction

Vocabularies can then be built as unions of elementary vocabularies, obtained through specialization criteria that make sense for a given informational purpose (see Section 4). The use of the same specialization criterion in the definition of different elementary vocabularies (for instance in Figure 3, vocabularies V_3 and V_5) can explain why categories that are at different depths in the ontology may be pertinent for the same uses.

3.2 An equivalent definition

The main idea being that the depth in the ontology is not so important as the specialization criterion, we propose to formalize the notion of criterion as a specialization of the “kind of” relation.

Definition 3 *A specialization of the “kind of” relation (noted $<_{crt}$) is a restriction of the “kind of” relation obtained by specifiying the criterion crt used to establish it.*

In Figure 3, 4 direct specializations of the “kind of” relation are used to define the elementary vocabularies: “kind of, with regard to role for humans” (noted koR), “kind of, with regard to composition” (noted koC), “kind of, with regard to appellation” (noted koA), “kind of, with regard to technological process” (noted koT). They could themselves be specialized, as proposed in Figure 4.

Elementary vocabularies can now be re-defined on the basis of the specializations of the “kind of” relation (more simply called: “kind of” relations, in the following) used to define them.

[4] declaratively defined.

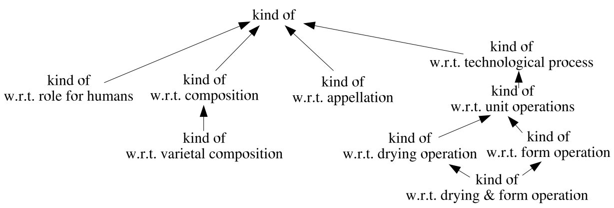

Figure 4. Specializations of the "kind of" relation

Definition 4 *Given: (i) a set of "kind of" relations, and (ii) a set of concept types T_C in which each pair (t, t'), where t' is a direct specialization of t, is associated with the "kind of" relation used to specialize t into t',*
an elementary vocabulary is a set of elements of T_C having the same alternation[5] *of "kind of" relations on their paths from* Universal.

For example, in Figure 3, there is one path from *Universal* to *Extruded pasta*, with the following "kind of" relations: koR, koC, koC, koC, koA, koT. The alternation of "kind of" relations on this path is thus: koR, koC, koA, koT. From *Universal* to *Dry laminated pasta*, there are 2 paths (one through *Laminated pasta* and one through *Dry pasta* that both have the same "kind of" relations: koR, koC, koC, koC, koA, koT, koT. The alternation of "kind of" relations on these paths is: koR, koC, koA, koT. As *Extruded pasta* and *Dry laminated pasta* have the same alternation of "kind of" relations on their paths from *Universal*, they belong to the same elementary vocabulary according to Definition 4.

Definitions 2 and 4 of an elementary vocabulary can be shown to be equivalent.

4 MAPPING CONTEXTS TO VOCABULARIES

4.1 A mapping between contexts and vocabularies

A vocabulary, built as unions of elementary vocabularies, makes sense for a given informational purpose, corresponding to a given context (nesting type). Hence we propose to associate a vocabulary with each nesting type.

Definition 5 *Each nesting type is associated with a vocabulary through a mapping noted v from the set of nesting types to the set of (non-elementary) vocabularies, satisfying: given two nesting types n and n', if n' is more specific than n then $v(n') \subseteq v(n)$.*

For example, the general nesting type *Description* can be associated with T_C. The vocabulary associated with the more specific nesting type *Nutritional description* excludes sanitary and biochemical elementary vocabularies (Sanitary quality, Phytosanitary content, Thermolabile vitamin, Hydrosoluble vitamin ...). The vocabulary associated with *Nutritional observation* excludes the evaluation elementary vocabulary (Improvement, Deterioration, Quality ...). This is illustrated by Figure 2.

4.2 Context-specific querying

The so-called "projection" mechanism of conceptual graphs, which is the basis of querying in that model, remains unchanged using this representation of vocabularies. This is due to the fact that the vocabulary associated with a nesting type (that appears in a query for instance) includes the vocabulary associated with a more specific nesting type (which can appear in an answer to this query), which avoids having answers whose vocabulary is unknown to the query.

Figure 5 gives an example of a query that expects answers (about food products) to be in the nutritional field. The conceptual graph of Figure 2 provides two answers, contained in the *Nutritional observation* and *Nutritional evaluation* nestings (these types are more specific than *Nutritional description* present in the query).

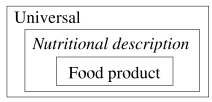

Figure 5. Example of nutrition-specific query

5 CONCLUSION AND PERSPECTIVES

This work has proposed two equivalent ways of defining vocabularies, the first one based on the identification of elementary vocabularies, the second one on specializations of the "kind of" relation. A mapping between contexts, represented as nesting types, and vocabularies has been proposed, which is in accordance with the querying mechanism of the conceptual graph model.

This work, emerging from user needs in an application in food science, should evolve in several directions. A first perspective is an extension in order to provide complementary answers during the querying, e.g. answers from other contexts – that is, from nestings with a non-comparable nesting type – that have compatible vocabularies (common concept types) and that effectively only use concepts that are allowed in the context of the query.

An important issue will be to give the user the choice of the "kind of" relations used in the querying, that can be different from one part of a query to another, so as to allow a rich expression of needs.

REFERENCES

[1] S. Buvač, 'Resolving lexical ambiguity using a formal theory of context', in *Semantic Ambiguity and Underspecification*, CSLI, (1996).
[2] M. Chein, M.L. Mugnier, and G. Simonet, 'Nested graphs: A graph-based knowledge representation model with fol semantics', in *KR'98*.
[3] J. Esch, 'Contexts and concepts, abstraction duals', in *Conceptual Structures: Current Practices*, 175–184, Springer-Verlag, (1994).
[4] A. Firat et al., 'Multi-dimensional ontology views via contexts in the ECOIN semantic interoperability framework', in *C&O-2005*.
[5] N. Guarino, 'Formal ontology and information systems', in *Proceedings of FOIS'98*, pp. 3–15, Trento, Italy, (June 1998). IOS Press.
[6] R.V. Guha, *Contexts: A Formalization and Some Applications*, Ph.D. dissertation, Stanford, 1991.
[7] G. Van Heijst, A.T. Schreiber, and B.J. Wielinga, 'Using explicit ontologies in KBS development', *IJHCS*, **46**, 183–292, (1997).
[8] P. De Leenheer and A. de Moor, 'Context-driven disambiguation in ontology elicitation', in *C&O-2005*.
[9] J. McCarthy, 'Generality in artificial intelligence', *Communication of the Association for Computing Machinery*, **30**(12), 1030–1035, (1987).
[10] G. Mineau and O. Gerbé, 'Contexts: A formal definition of worlds of assertions', in *Proceedings of ICCS'1997, LNAI#1257*, Seattle, (1997).
[11] J.F. Sowa, *Conceptual structures - Information processing in Mind and Machine*, Addison-Welsey, 1984.
[12] V. Terziyan and S. Puuronen, 'Reasoning with multilevel contexts in semantic metanetwork', in *Formal Aspects in Context*, Kluwer, (2000).

[5] i.e. if the same 'kind of" relation appears several times consecutively in the path, it is considered only once

Combining Contexts and Ontologies: A Case Study and a Conceptual Proposal

Mariela Rico [1] and **Ma. Laura Caliusco** [2] and **Omar Chiotti** [1, 3] and **Ma. Rosa Galli** [1, 3]

Abstract. Recently, approaches that combine contexts and ontologies taking advantages of their strengths have been developed. Each of them solves different problems from different perspectives. The objective of this paper is to present problems that are not solved up to date as well as to introduce a conceptual proposal. To this aim, we base our analysis on a collaborative B2B scenario that is relevant for todays competitive and highly dynamic environment.

1 INTRODUCTION

Today, there is an increasing interest on combining context and ontology to define information semantics. The effort for combining both approaches could be classified according to the objectives to be achieved in works focused on: modelling and knowledge representation [1], [2]; and achieving interoperable systems that require data from multiple information sources [3], [4]. In the first case, an ontology alignment that consists on defining different kinds of relations between the involved ontologies, is enough to achieve semantic interoperability. In the other, however, it is crucial to define an ontology mapping that consists on defining equivalence relations [5].

Particularly, our research is focused on achieving semantic interoperability between heterogeneous information systems to support a collaborative business-to-business (B2B) relation between trading partners. In this area, the main approaches are focused on the idea of similar ontologies. However, each enterprise has its own information systems, and the challenge here is how to semantically integrate these heterogeneous systems.

The objective of this paper is to present problems that are not solved up to now and introduce a conceptual proposal for combining context and ontology. To this aim, the paper is organized as follows: Section 2 presents related works. Section 3 presents a context definition. Section 4 presents a case study based on a collaborative B2B scenario, and discusses problems that arise when combining contexts and ontologies. Section 5 presents conclusions and future work.

2 RELATED WORKS AND DISCUSSION

It is possible to find formal and informal approaches defining an ontology [5], and a context [6], [7], [8]. Ontologies define a common understanding of specific terms, and thus make it possible the semantic interoperability between systems, but they can only be used after reaching consensus about their content. Contexts encode not shared individual interpretation schemas, that are easy to define and maintain since they can be created with a limited consensus among parties, but the communication between systems can be achieved only by constructing explicit mappings [4].

Therefore, taking into account that the strengths of ontologies are the weaknesses of contexts and vice versa, a number of approaches were developed which propose to combine both concepts to achieve information semantic interoperability. Following, we analyze two of them that are recent and improve others previously defined in the area.

2.1 A centralized approach

In [3], the ECOIN (Extended COntext INterchange) semantic interoperability framework has been defined. It proposes to define a single ontology consisting of generic terms without specifying their exact semantics and it specializes them in local contexts to express specific meanings. ECOIN defines mappings structuring lifting axioms [7] as a conversion function network, defining them for each modification dimension according to a context model. ECOIN results in a simpler context model, which works very well in a domain where it is possible to define a single ontology and to relate it with multiple contexts.

2.2 A decentralized approach

In [4], an extension of the OWL language, C-OWL, has been defined to represent contextual ontologies where a context is a concrete domain viewed from the description logic perspective. In this work, ontology is contextualized when its contents are kept local and mapped with the contents of other ontologies via explicit mappings using bridge rules. These rules represent the relations: *equivalent to, more general than, less general than, compatible and incompatible.* C-OWL allows a user to define ontologies alignment where it is inappropriate to define a global shared ontology. However, the limited expressiveness of the C-OWL fails to address the contextual differences found in most practical settings, as it will be shown later.

3 OUR VIEW OF CONTEXT

When we talk about context, intuitively, we think in the set of facts in which something exits or occurs. This idea is not reflected by approaches described in Section 2. Our intention is to apply the theory about context [7], [8], for information semantics modelling and combine it with ontology. In our opinion this is a more appropriate way to take advantage of both approaches strengths in complex domains.

In the theory defined in [7] and [8], axioms and statements p are only true in a context c. This fact is expressed by the formula

$$c' : ist(c; p)$$

[1] CIDISI-UTN-FRSF, Argentina, email: mrico@frsf.utn.edu.ar
[2] CIDISI-UTN-CONICET, Argentina, email: mcaliusc@frsf.utn.edu.ar
[3] INGAR-CONICET, Argentina, email: chiotti,mrgalli@ceride.gov.ar

The context c is formalized as a first class object. Formulas *ist(c,p)* are always considered as themselves asserted within a context, *c'*. Although this formulas define a infinite regress, to manipulate context we have to define a limit.

In information semantics modelling area we could state that a context is a set of facts in which a concept interpretation is true.

$$c = \{a\ set\ of\ facts\} \quad and \quad p = concept\ interpretation$$

c and p could be an ontology or an ontology element.

Defining contexts as a set of facts instead of a label allows us to manipulate them in a flexible way as will be shown next.

About contexts relations, there are two proposed ways: lifting axioms and bridge rules [12]. The main difference between them is that lifting axioms are stated in an external context, which must be expressive enough to represent facts of all involved contexts, whereas bridge rules allow stating relations between contexts without the need of an external one. In Section 4.3, advantages of having an external context are analyzed. So, a relation *R* between context will be

$$c' : (ist(c_1; p_1) \quad \xrightarrow{R} \quad ist(c_2; p_2))$$

4 A CASE STUDY

Nowadays, enterprises work together with their trading partners to improve supply chain (SC) management. Then, there is a semantic heterogeneity that could be solved by using ontologies, but it is not enough. Two concepts can be differently related to each other in different contexts, as different enterprises. In a collaborative relation, it is primordial that each enterprise preserves its identity, particularly the semantic identity. So, it is necessary to make the context explicit.

From a business point of view, to allow a decentralized management, the *PartnertoPartner Collaborative Model* has been defined in [10], which proposes a peer-to-peer collaboration between trading partners. In this model, decisions are independently made with the aim of preserving the privacy and autonomy of each enterprise. Let us define an example where a brewery has collaborative relations with two of its clients: a retailer and a warehouse; each relation constitutes a different context, C_{RBR} and C_{RBW} respectively.

The management of each collaborative relation implies coordinating: private processes (PP) that are executed by each enterprise; and collaborative processes (CP) that are jointly executed by trading partners. CP are defined as abstract ones; and in order to implement it each trading partner has to define a business interface process (IP). This IP is responsible for the invocation and execution of those PP required for carrying it out. To allow the CP execution, Electronic Business Documents (EBDs) are exchanged between trading partners. EBDs are standardized data structures that replace traditional business documents [11].

When the IP receives an EBD, it has to translate it information to the PP according to the semantic of corresponding enterprise sector. Then, to send an EBD, the IP populates it with data of corresponding enterprise sector according to the CP semantics. So, to make interoperable systems, the IP has to solve a number of conflictive situations at semantic level considering that it is necessary to define equivalence relations between concepts. Next, some of them are analyzed from the theories defined in the previous sections considering the relation between the brewery and a retailer.

4.1 Multiple ontologies and multiple domains

In a collaborative relation, each EBD is described by an ontology [11] that is not a global or general one, but only an ontology that describes the EBD information semantics, which was agreed by both partners. The EBDs will be processed by partners' private processes that may involve different enterprise sectors. Even two sectors within the same enterprise for apparently similar applications have different views, resulting in similar but still not the same ontology. Then, it is clear that each enterprise has its own ontologies to describe the semantics of its systems and internal areas.

Figure 1.b shows a part of an ontology (O_{EBD}) shared by both trading partners that describes the semantics of EBD interchanged to agree on a replenishment plan. Even if both supplier and client could have multiple ontologies to describe their information semantics, in this paper and for simplicity purposes, we focus on one ontology for each enterprise, O_S and O_C respectively, Figure 1.a and 1.c.

Although the centralized integration proposal (Section 2.1) introduces a simpler ontological model, it is difficult and sometimes even impossible to implement this proposal in a collaborative B2B relation. That is because each enterprise in a SC has its own interests; and its information systems and data structures have been designed to achieve those interests. So, when these enterprises decide to join themselves in a collaborative process, they do it with a common interest but keeping their individuality and privacy. As regards context, the ECOIN definition is not applicable either, since a context is more than just possible instance values in a collaborative B2B scenario.

The decentralized proposal (Section 2.2), seems to be more appropriate to model this scenario, since it handles different ontologies. However, the manipulation of contexts lacks of needed expressively to represent that within the context C_{RBR} exists subcontexts such as Supplier, C_S; Collaborative Process, C_{CP}; and Client, C_C. This fact can be modelled with the theory of context described in Section 3 as:

$$C_{RBR} \quad : \quad ist(C_S, p_S)$$

$$C_{RBR} \quad : \quad ist(C_{CP}, p_{EBD})$$

$$C_{RBR} \quad : \quad ist(C_C, p_C)$$

where p_S, p_{EBD}, and p_C are truth propositions in their contexts.

4.2 Contexts within an ontology

Considering the *Type* term of the O_{EBD} ontology (Figure 1.b), it is associated to the *Packaging* and *Product* terms. Even though *Type* has the same semantics, since it describes the class or nature of the concepts it is associated to, the possible values it may take are different. In the case of *Packaging*, *Type* can be *Can* or *Bottle*. But, for *Product*, *Type* can be *Local* or *NKH*. This presents an ambiguity problem that could be solved by replacing the term *Type* by *PackagingType* and *ProductType*. In this way, however, terms are unnecessarily added to the ontology, and this practice could lead to a size increase [8]. In our opinion, a better solution is to consider *Product* and *Packaging* as different contexts inside of which the term *Type* is interpreted,

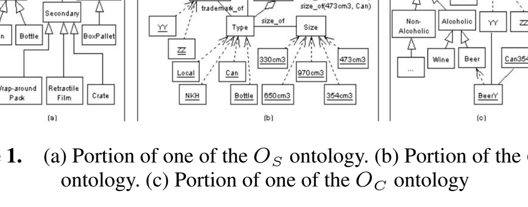

Figure 1. (a) Portion of one of the O_S ontology. (b) Portion of the O_{EBD} ontology. (c) Portion of one of the O_C ontology

defining them by a set of formulas like one shown in Table 1. In this table, *Product* and *Packaging* refer to O_{EBD} terms, and they are not simple labels, which give name to the contexts.

Table 1. Definitions of *ProductCxt* and *PackagingCxt* contexts

ProductCxt context	PackagingCxt context
ist(Product, part_of(Product, Type))	ist(Packaging, part_of(Packaging, Type))
ist(Product, Type(Local))	ist(Packaging, Type(Can))
ist(Product, Type(NKH))...	ist(Packaging, Type(Bottle))...

Here, different contexts are created within an ontology with the aim of solving name ambiguities [8]. This problem has no been tackled in the literature related to information system interoperability.

4.3 Relating contexts

Figure 1.c shows a client ontology portion, O_C. If the term *Trademark* is considered, it is an attribute of *Product* in O_C. This means, it is an attribute of all products and not just of beer. By contrast, *Trademark* $\in O_{EBD}$ (Figure 1.b) has a *part_of* relation with *Product*. Furthermore, *Trademark* $\in O_{EBD}$ has an association with *Type*, which is valid in the context *ProductCxt* but not on the context *PackagingCxt*. This relation does not exist in O_C because this information is irrelevant for the client. In spite of these differences, however, we can say that *Trademark* $\in O_{EBD}$ is equivalent to *Trademark* $\in O_C$ since their instances are equivalent to the collaborative context. These terms could be related by using equivalence mapping rules [4]:

$$O_{EBD} : Trademark \xrightarrow{\equiv} O_C : Trademark \quad \wedge$$

$$O_C : Trademark \xrightarrow{\equiv} O_{EBD} : Trademark$$

Considering this example, mapping rules defined in [4] are useful in cases where simple equivalence relations are enough to express similarities between contexts.

However, if previous rules are analyzed from the client context C_C point of view, these relations are not truth. The term *Trademark* of O_C is more general than the term *Trademark* of O_{EBD}, since it represents the trademark of all products and not only beers. That is:

$$O_{EBD} : Trademark \xrightarrow{\subseteq} O_C : Trademark$$

Previous rules, defined in this way, could carry incompatibility problems. A possible solution should be to contextualize them:

$$ist(C_{CP}, (O_{EBD} : Trademark \xrightarrow{\equiv} O_C : Trademark))$$

$$ist(C_C, (O_{EBD} : Trademark \xrightarrow{\subseteq} O_C : Trademark))$$

It is necessary to clarify that this is not a formalization, but only a way to express the idea that the rules linking terms belonging to different concepts also should be contextualized.

4.4 Different contexts, different representations

By comparing O_C and O_{EBD} (Figure 1.b - 1.c), the concept represented by *PackageType* in O_C is equivalent to *Size* and *Type* terms in O_{EBD}, for *Type* in the *PackagingCxt* context, but not in the *ProductCxt* context. So, *PackageType* $\in O_C$ is related to *Packaging*, *Type* and *Size* $\in O_{EBD}$ plus their relations. Analyzing the instances, *PackagingType(Can354cm3)* has to be translated into O_{EBD} as:

$$\begin{aligned} O_{EBD} \quad : \quad & ist(Packaging, (Type(Can)) \wedge \\ & ist(Packaging, Size(354cm3)) \wedge \\ & ist(Packaging, size\_of(354cm3, Can)) \wedge \\ & ist(Packaging, part\_of(Type(Can), Packaging)) \end{aligned}$$

That means that a certain concept is represented by a term in a particular ontology, but is represented as a set of terms, a set of relations and a context in another ontology. This example shows that in order to define mappings between different contexts it is necessary to define conversion rules that are more complex than mapping rules defined by [4] and the conversion function defined by [3].

5 CONCLUSIONS AND FUTURE WORK

The main contribution of this paper is a conceptual proposal that combines contexts and ontologies in order to manipulate semantic differences in a complex domain, such as a collaborative B2B scenario. This proposal is based on a previously defined context theory, however, we have explored the possibility to combine it with ontology concepts. Our approach proposes to define contexts as a set of facts that allow us to manipulate it in a more flexible way.

In a complex domain, having an external context may be an advantage. So, an interesting option to be analyzed is the definition of lifting axioms to define conversion rules between contexts. This analysis will be the focus of our future work, however, in this paper we have made progress in this sense. An important feature of these conversion rules is they have to allow us to relate a term in an ontology with a set of terms, a set of relations and a context in another.

The present proposal is incomplete and tentative since this is just the first step and further research remains to be done.

REFERENCES

[1] A. Segev, and A. Gal, *Putting Things in Context: a Topological Approach to Mapping Contexts and Ontologies*, 9–16, Contexts and Ontologies: Theory, Practice and Applications, AAAI Press, 2005.

[2] P. De Leenheer, and A. de Moor, *Context-driven Disambiguation in Ontology Elicitation*, 17–24, Contexts and Ontologies: Theory, Practice and Applications, AAAI Press, 2005.

[3] A. Firat, S. Madnick, and F. Manola, *Multi-dimensional Ontology Views via Contexts in the ECOIN Semantic Interoperability Framework*, 1–8, Contexts and Ontologies: Theory, Practice and Applications, AAAI Press, 2005.

[4] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt, 'Contextualizing Ontologies', *Web Semantics: Science, Services and Agents on the World Wide Web*, **1**: 4, 321–419, (2004).

[5] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering*, Springer Verlag, London, 2nd edn., 2004.

[6] P. Brzillon, 'Context in problem solving: A survey', *The Knowledge Engineering Review*, **14**: 1, 47–80, (1999).

[7] R. Guha, *Contexts: A Formalization and Some Applications*. PhD Thesis. Computer Science Dep., Standford University, 1995.

[8] J. McCarthy, and S. Buvac, *Formalizing Context* (Expanded Notes), Computer Natural Language, 1997.

[9] M. Theodorakis, *Contextualization: An Abstraction Mechanism for Information Modeling*. PhD Thesis. University of Crete, Greece, 2001.

[10] P. Villarreal, M. Caliusco, M. Galli, E. Salomone, and O. Chiotti, *Decentralized Process Management for Inter-Enterprise Collaboration*, 69–79, Vision, Special Issue on Supply Chain Mgment, **7**, India, 2003.

[11] M.L. Caliusco, M. R. Galli, and O. Chiotti, 'Ontology and XML-based Specifications for Collaborative B2B Relationships', *CLEI Electronic Journal*, **7**: 1, (2004).

[12] P. Bouquet, and L. Serafini, *On the difference between bridge rules and lifting axioms*, 80–93, Proc. 4th CONTEXT, LNCS 2680, 2003.

Ontology Verification Using Contexts

Aviv Segev and **Avigdor Gal**
Technion – Israel Institute of Technology
{asegev@tx, avigal@ie}.technion.ac.il

Abstract. Ontologies have become the de-facto modeling tool of choice, used in a variety of applications and prominently in the Semantic Web. Their design and maintenance, nevertheless, have been and still are a daunting task. As a result, ontologies quickly become underspecified. Therefore, if ontologies do not evolve, the semantic infrastructure of the information system can no longer support the changing needs of the organization. In this work we provide a model to semi-automatically support relationship evolution in an ontology using contexts. We propose to use (machine-generated) contexts as a mechanism for quantifying relationships among concepts. To do so we compare the contexts that are associated with the ontology constructs. On a conceptual level, we introduce an *ontology verification* model, a quantified model for automatically assessing the validity of relationships in an ontology. We motivate our work with examples from the field of eGovernment applications. To support our model with an empirical analysis, we provide a mapping of an ontology operator for defining relationships into context relationships, using real-world traces of RSS.

1 INTRODUCTION

Ontologies have become the de-facto modeling tool of choice, used in a variety of applications and prominently in Semantic Web applications. For example, ontologies can be used in discovering Web services [10]. Ontology design, nevertheless, has been and still is a daunting task. It requires collaboration of domain experts with ontology engineers, which may consume many organizational resources in terms of both time and monetary units. Once the ontology is designed, evolving it becomes difficult due to the need for availability of domain experts on the one hand, and costs related with hiring ontology engineers on the other hand. To illustrate this point, consider an eGovernment application, for which an ontology was designed and tailored by an ontology engineer. Once the ontology is installed, changes in the real world require a renewed collaboration of civil servants with ontology engineers to reflect such changes in the ontology. A typical outcome of such difficulties is that ontologies quickly become underspecified. New concepts are introduced in the domain while others become obsolete. Also, shifts of focus in the application domain require the refinement of a concept into a hierarchy of concepts, while in other cases hierarchies should be collapsed. Meeting these challenges requires ontologies to evolve or else the semantic infrastructure of the information system can no longer support the changing needs of the organization.

In [9] we introduced a model for compensating for ontology underspecification using a combination of ontologies with contexts. *Contexts* were defined to be first class objects [5] and will be formally presented later in this work. As an example, a context can be defined to be a set of words, possibly associated with weights that represent the relevance of a word to a document. Ontologies and contexts are both used to model different perspectives of a domain (views). Ontologies represent shared models of a domain and contexts are local views of a domain. We also promote an orthogonal classification in which ontologies are considered a result of a manual effort of modeling a domain, while contexts are system generated models [8]. Ontologies and contexts are joined together, as formally described in [9]. In a nutshell, each concept in an ontology is represented by a name and a context. In this model, contexts serve as an easy-to-use "semantic glue," in which underspecifications are compensated for with a syntactic, machine generated context, which highlights the intentions of a local designer when using a specific ontology concept, possibly differently from the way it is semantically captured in the ontology using relationships.

In this work we provide a model and an example of an algorithm to semi-automatically support relationship evolution in an ontology using contexts. The main motivation for this work stems from the difficulty in supporting ontology evolution. Specifically, this problem was raised within the framework of TerreGov, a European eGovernment project. In this project, ontologies serve as the driving force behind the application and thus affect government processes and Web services, among other things. Therefore, we propose to use (machine-generated) contexts as a mechanism for quantifying relationships among concepts. Specifically, given an ontology operator (*e.g.*, *link subclass*, representing the knowledge that an instance of one concept is included in an instance of another) and operands (*e.g.*, two concepts or classes), we aim at quantifying the extent to which this relationship is valid. We do so by comparing the contexts that are associated with the operands. We believe that such a solution would significantly assist in the support of ontology design and evolution.

The main contribution of this work is thus twofold. On a conceptual level, we introduce an *ontology verification* model, a quantified model for automatically assessing the validity of relationships in an ontology. On an algorithmic level, we provide an example of a mapping of ontology operator for defining relationships into context relationships. We motivate our work with examples from the eGovernment domain. However, due to the absence of large scale data sets for this domain, we support our model with an empirical analysis using real-world traces of RSS data.

The rest of the paper is organized as follows. We start with preliminaries, formally defining ontologies and contexts in Section 2. In Section 3 we introduce the ontology verification model, followed by an example of a mapping of the ontology verification problem to contexts in Section 4. We conclude with related work in Section 5 and a short summary in Section 6.

2 ONTOLOGIES AND CONTEXTS

Banerjee [1] defined a *root class* as an object that represents anything from a simple number to a complex entity. An edge between a node and a child node in a class represents an IS-A relationship. Objects that belong to a class are called *instances* of that class. A class describes the *form* (instance variables) of its instances and the *operations* (methods) applicable to its instances.

According to Gruber [2], an *ontology* is an explicit specification of a domain conceptualization. Several models for ontologies exist; we follow here that presented in [2]. In the discussion below, we assume reader familiarity with basic concepts in conceptual modeling.

We define a *context* $\mathcal{C} = \left\{\{\langle c_{ij}, w_{ij}\rangle\}_j\right\}_i$ as a set of finite sets of descriptors c_{ij} from a domain $\mathcal{D}$ with appropriate weights w_{ij}, representing the importance of c_{ij}. For example, a context $\mathcal{C}$ may be a set of words (hence, $\mathcal{D}$ is a set of all possible character combinations) defining a document Doc, and the weights could represent the relevance of a descriptor to Doc. In classic Information Retrieval, $\langle c_{ij}, w_{ij}\rangle$ may represent the fact that the word c_{ij} is repeated w_{ij} times in Doc.

The context of a class is defined as a set of contexts describing instances that belong to this class. Documents are not instances but represent them. Following [9], we define a class context $\mathcal{C}_{CL}$ of a class CL to be the union of its instance contexts.

Segev and Gal [9] aimed at formalizing the inter-relationships between an ontology, a manually generated domain model, and contexts, partial and automatically generated local views. According to their work, a context can belong to multiple context sets, which in turn can converge to different ontology concepts. Thus, one context can belong to several ontology concepts simultaneously. The appropriate interpretation of a context leads to its relevance to different given concepts.

3 ONTOLOGY VERIFICATION USING CONTEXTS

Ontology verification is the process by which semantic relationships are identified. We term this process verification, since we assume an ontology exists and may need to evolve. Therefore, semantic relationships in an ontology need to be continuously monitored and if necessary, revised. Here we follow the work of [6] on ontology changes and assume a given closed set of operators OT, to be applied on a set of operands OD, taken from the set of all ontology elements. As an example, a change operator may be the *disjoint* operator, resulting in the creation of a semantic relationship called "disjoint" between two classes, given to it as operands.

Figure 1 provides a pictorial representation of the process. Formally, ontology verification is a function $OV : OT \times OD^* \rightarrow [0, 1]$. Ontology verification is given as input a hypothesis regarding the possible operator to be applied to one or more operands and returns a level of certainty μ regarding the truth in this hypothesis. A certainty of 1 indicates full certainty in the hypothesis, while a certainty of 0 means that the hypothesis is definitely incorrect. In Figure 1, the ontology verification function determines that the disjointedness of classes CL_1 and CL_2 has a certainty level of 0.9. An example of the use of the model can be a user who would like to analyze a local government concept relationship. The user could supply a set of documents representing two concepts and could receive a verification level based on this representative set of documents.

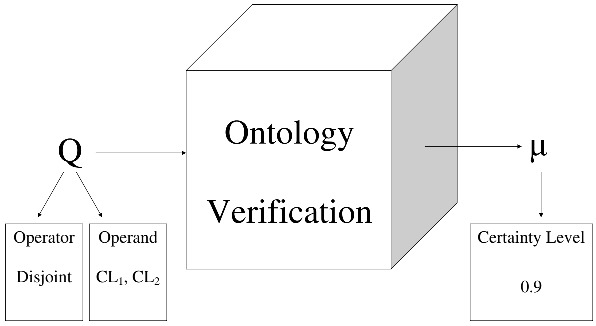

Figure 1. Ontology Verification Model

4 EXPERIENCES WITH CONTEXT BASED ONTOLOGY VERIFICATION

Having introduced ontology verification, we now focus on the details of change operators. Noy and Klein [6] describe a set of 22 ontology change operators and their impact on ontology elements (both classes and instances) using ontology relations defined in [2]. We take one of their ontology change operators and use it as an example.

Our experiences are based on data from the RSS news data trace. In this data trace, data were originally partitioned to topics with no ontological relationships. The RSS trace was collected during August 2005 from the CNN Web site. We chose 10 news topic categories for the data, when each RSS news header or news descriptor constitutes a datum. We generated a context for each datum and each class using an automatic context extraction algorithm [8]. The number of context descriptors generated from each datum was set to 10. The data size used for RSS varied from 73 to 1,911 per class.

In our experiment we calculated for each class the number of contexts that overlapped with the other nine classes. This asymmetric comparison gave us for any two classes CL_i and CL_j the metric of $\left|\mathcal{C}_{CL_i} \cap \mathcal{C}_{CL_j}\right|$ and $\left|\mathcal{C}_{CL_i} \cup \mathcal{C}_{CL_j}\right|$.

Given two classes, CL_i and CL_j, if CL_i is a subclass of CL_j, then its context should be contained in the context of CL_j. This is because an instance of CL_i is also an instance of CL_j and therefore has a broader context than an instance of the superclass. Therefore, we compute the certainty of a hypothesis that CL_i is a subclass of CL_j to be

$$\mu_{\textit{Sub-Sup}} = \frac{\left|\mathcal{C}_{CL_i} \cap \mathcal{C}_{CL_j}\right|}{\left|\mathcal{C}_{CL_j}\right|}$$

Our experience involves an analysis of hierarchy linking. Figure 2 presents the RSS class relations hierarchy created for $\mu_{\textit{Sub-Sup}} \geq 0.8$ and $\mu_{\textit{Sub-Sup}} \geq 0.5$. As the value of $\mu_{\textit{Sub-Sup}}$ decreases, the hierarchy and the relations between the classes become more elaborated. For example, in the RSS data for $\mu_{\textit{Sub-Sup}} \geq 0.8$ the superclass Money Latest has four subclasses. If we examine the same classes for a lower verification level of $\mu_{\textit{Sub-Sup}} \geq 0.5$ we receive a three level hierarchy.

Table 1 compares the certainty level of the *Superclass-Subclasss* operator, for two class pairs in the RSS data set. When evaluating the classes Money Latest and Money News International, there is a high $\mu_{\textit{Sub-Sup}}$ level.

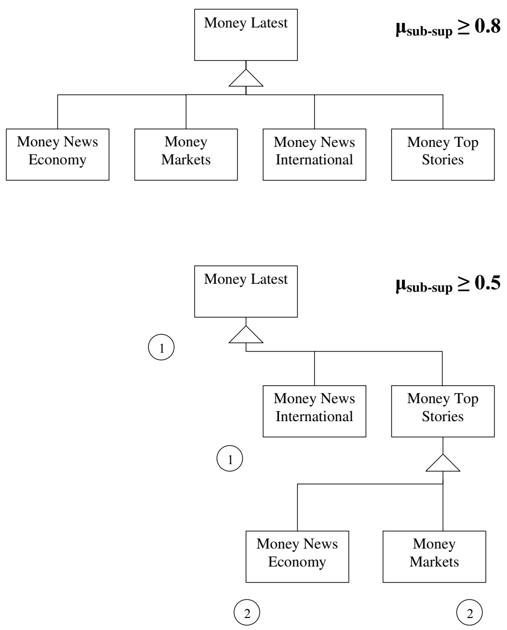

Figure 2. RSS Relations

5 RELATED WORK

A formal mathematical framework that delineates the relationships between contexts and ontologies is presented in [9]. To deal with the uncertainty associated with automatic context extraction from existing instances, such as documents, a ranking model was provided, which ranks ontology concepts according to their suitability with a given context.

A semi-automated method for ontology evolution using documents clustering was proposed in [11]. From the results of the clustering ontology enrichments and updates are extracted. In contrast to the above work, which is based on a single word ontology concept description, we use a set of contexts describing each ontology class.

Noy and Klein [6] defined a set of ontology-change operations and their effects on instance data used during the ontology evolution process. They describe ontologies schemas and database schemas from the point of view of evolution and highlight the main differences between them. We presented a model that shows how these ontology change operations can be verified based on context.

Tools for merging and aligning ontologies, such as SENSUS [3], PROMPT [7], and Cyc [4], have been developed in the past. These tools generally present a set of basic operations that are performed during the mergence and alignment of ontologies and that determine the effects that each of these operations has on the process.

A work on multi-contextual ontology evolution [12] defines a set of properties that by semantic autonomy must hold at the same time.

6 CONCLUSION

This work presents a model and a set of algorithms to semi-automatically support ontology relationship evolution using contexts.

Class Sets	Link Subclass
Money Latest	86.7%
Money News International	19.8%
Money News Economy	19.5%
Money Markets	24.3%

Table 1. Operator μ Verification RSS

Given an ontology operator and operands, the model provides the quantification of the extent to which the relationship is valid. The model is supported by empirical analysis, using initial experiences with real-world RSS traces. The experiences with these traces show how relationships between the classes can be created and modified. Preliminary empirical results show that our model can provide good estimations of the need for ontology changes.

To recap, the main contribution of this work is both conceptual and algorithmic. We present an ontology verification model, a quantified model for automatically assessing the validity of relationships in an ontology, and we also provide a mapping of several ontology operators for determining relationships among classes.

The results of this work will be embedded as part of the TerreGov solution. Future research will examine the model performance on eGovernment data and other large data sets. In addition, we plan on extending the model to include additional operators.

REFERENCES

[1] J. Banerjee, H.-T. Chou, J. Garza, W. Kim, D. Woelk, and N. Ballou, 'Data model issues for object-oriented applications', *ACM Transactions on Office Information Systems*, **5**(1), 3–26, (1987).

[2] T. R. Gruber, 'A translation approach to portable ontologies', *Knowledge Acquisition*, **5**(2), (1993).

[3] K. Knight and S. K. Luk, 'Building a large-scale knowledge base for machine translation', in *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, (1994).

[4] D. B. Lenat, 'Cyc: A large-scale investment in knowledge infrastructure', *Communications of ACM*, **38**(11), 33–38, (1995).

[5] J. McCarthy, 'Notes on formalizing context', *In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, (1993).

[6] N. F. Noy and M. Klein, 'Ontology evolution: Not the same as schema evolution', *Knowledge and Information Systems*, **6**(4), 428–440, (2004).

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

[8] A. Segev, 'Identifying the multiple contexts of a situation', in *Proceedings of IJCAI-Workshop Modeling and Retrieval of Context (MRC2005)*, (2005).

[9] A. Segev and A. Gal, 'Putting things in context: A topological approach to mapping contexts and ontologies', in *Proceedings of AAAI-Workshop Workshop on Contexts and Ontologies: Theory, Practice and Applications*, (2005).

[10] E. Toch, A. Gal, and D. Dori, 'Automatically grounding semantically-enriched conceptual models to concrete web services', in *ER*, eds., L.M.L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, and O. Pastor, volume 3716 of *Lecture Notes in Computer Science*, pp. 304–319. Springer, (2005).

[11] G. Tsatsaronis, R. Pitkanen, and M. Vazirgiannis, 'Clustering for ontology evolution', in *Proceedings of the 29th Annual Conference of the German Classification Society (GfKl 2005)*, (2005).

[12] M. Zurawski, 'Reasoning about multi-contextual ontology evolution', in *Proceedings of the First International Workshop on Context and Ontologies: Theories, Practice and Applications, The Twentieth National Conference on Artificial Intelligence (AAAI-05)*, (2005).

Mapping ontologies and contexts: from theory to a case study

Dmitry Kudryavtsev[1]

Abstract: Ontologies are used actively as a knowledge representation, retrieval and navigation tool to improve knowledge sharing, exchange and communication. In order to provide effective communication ontologies should be mapped with the context. This paper analyses existing approaches towards the very definition of context and suggests two context types. Requirements for effective knowledge representation based on two context types and on mapping ontologies and context are suggested. These requirements are considered and factored in the following case study by consecutive mapping different context types and content ontology . This case study describes Knowledge Navigator – a map that relates contents of Formalized Management methodology with the corresponding context in order to reach effective knowledge communication to end users.

1 INTRODUCTION.

Nowadays organizations implement special tools and technologies to share, exchange and communicate knowledge. In order to be effective, these tools and technologies must provide users with relevant information in due time without being flooded with irrelevant data. To support the sharing and exchange of knowledge both among information systems and people it is useful to define ontology [6]. Now ontologies are already employed in portals, corporate memories, e-commerce and other knowledge management systems (see [1], [2], [11]). With respect to human-computer interactions ontology often works as a representation, retrieval and navigation tool. In playing such a role ontology usually specifies the Content of knowledge resources. Such an ontology can be called Content ontology.

There are two problems that render the usage of Content ontology less efficient.

1. A Content ontology user is unable to set links between his/her task, problem, situation and notions in the Content ontology, thus he/she is unable to transform information into action.

"In many situations a content ontology user may not know the details of a solution, but he knows the details of his problem" [2].

"One of the fundamental tenets of knowledge management is that knowledge must link to and improve business processes. Without a map of the processes, goals, and knowledge assets inside one's organization, it will be difficult to reach one's destination." [14]

2. A Content ontology user is unable to match his/her personal mental model with notions in the Content ontology because of semantic and syntactical specialties of a person and ontology-creator.

This problem is taken from an elaborated field of semantic web where it is known as a mismatch between ontologies (see [5], [9]) (it is suggested to use analogy between ontology and personal mental model in the paper).

All these problems are related with the notion "context". These problems make problematic effective knowledge sharing and communication. In order to solve these problems it is necessary to define context and make explicit mapping between content ontology (or knowledge resource directly) and context. In the paper [Section 2] describes existent approaches to a context definition and mapping context and ontologies. [Section 3] marks out two context types and suggests the requirements for effective knowledge representation with respect to these types. [Section 4] initiates case-study and describes real-life knowledge communication task and corresponding problem. Knowledge Navigator (KN) is suggested as the solution for this knowledge communication task. [Section 5] suggests KN framework and brief description, which satisfies requirements from [Section 3] and is based on a consecutive mapping different context types and content ontology.

2 DEFINITION OF CONTEXT AND RELATED WORK

In [1] it is suggested to focus on the context as highly relevant for retrieval within an organization. In modeling the context the authors deal with two issues:

- the intended application context of a knowledge item, and
- the context a knowledge item was created in.

The Authors suggest that information context be expressed in terms of the *organizational structure and the process models*. These in turn are expressed in terms of the *enterprise ontology*. The design of the enterprise ontology is built on insights and developments from the enterprise modeling, business process modeling, and organizational modeling in knowledge-based systems [13]. In [2] a similar approach is used for semantic mapping between the sellers' supply and buyers' needs at an electronic knowledge market.

Similar enterprise ontology oriented approach to the context definition can be also found in the knowledge mapping technologies [4], [14].

The definition of context described above resulted from the knowledge management field, whereas in the semantic web field there is another useful definition of the context.

According to [3] *Ontologies* are *shared* models of some domain that encode a view which is common to a set of different parties

Contexts are *local* (where *local* is intended here to imply *not shared*) models that encode a party's view of a domain [7].

The authors argue that an ontology is contextualized, or that it is a *contextual ontology*, if it is kept local (and therefore not shared with other ontologies) but its contents are put in relation with the contents of other ontologies via explicit mappings. This mapping provides syntactic and semantic interoperability and deploys a variety of methods, coming from very different areas. They include: linguistic, statistical, structural and logical methods (see [5], [8], [9]).

Saint-Petersburg State Polytechnic University, Russia
BIG-Petersburg, Russia, dk@big.spb.ru

3 MODEL OF CONTEXTS AND REQUIREMENTS FOR EFFECTIVE KNOWLEDGE REPRESENTATION

Resuming Section 2 there are two main definitions of context that affect communication problems (Section 1):
Def 1. Context model reflects:

- the intended application context of a knowledge item, and
- the context a knowledge item was created in.

and is expressed in terms of the *enterprise ontology*.
Def 2. Contexts are *local (not shared)* models that encode a party's view of a domain.
In order to distinguish types of context and set requirements for knowledge representation working definitions are suggested for every type of context. The first working definition is based on a semiotic model [10]. Traditionally the semiotic model includes:

- Syntax which reflects rules and relations between signs of any language
- Semantics which reflect relations between signs and their meaning
- Pragmatics which reflect relations between signs and their users and creators

This model together with Def 1 makes possible to consider the context in Def 1 as *pragmatic context*.
The Context in Def 2 will cover all the components of the semiotic model making it impossible to define it uniformly in terms of a semiotic model. Thus the context in Def 2 will be termed and used in this paper as *local context.*
Pragmatic context can be either shared or not. Consequently the former is represented by ontology and the latter is by a set of local contexts.
The requirements for effective knowledge representation which provides for a solution of the problems from Section 1 are as follows:
Requirement 1: Every ontology must be either shared by *all* the communication participants *or* be mapped with corresponding local contexts of every participant (group of similar participants).
Requirement 2: Every knowledge resource must be mapped with a pragmatic context (either directly or by means of the content ontology).
These requirements are further considered and factored in the case study.

4 CASE STUDY: TASK SETTING AND PROBLEM DESCRIPTION

Formalized management methodology ("methodology" further in the paper) is a product of the management consulting company BIG-Petersburg. This methodology is initially presented in the form of a book, but the concept "Formalized Management Methodology" is used due to the plans of application of other media, such as e-books or knowledge portals.
This methodology reflects the experience of consultants gained during business process improvement and restructuring of organizations in Russia and CIS countries.
The goal of this methodology is to help different organizations in solving their managerial problems and improving levels of management. Thus the main objective is to provide each potential organization based user of the methodology with necessary knowledge to help realize the tasks and functions they face.
In order to achieve this objective the methodology must be effectively communicated to its potential users. Although methodology is well-structured with a content ontology (=table of contents) and divided into topics (content ontology nodes) it is rather hard to communicate it because the way the methodology can be used, its potential users and the methodology itself have their own specialties. These specialties can be considered as communication problem and are as follows:

a. Different organizations that intend to use the methodology face different problems and tasks. Many problems and tasks do not require usage of every topic of methodology.
b. Implementing such a methodology is not a task faced by one person or a small group only; it requires a joint effort made by many persons employed in the organization. As a result the target audience for the methodology implemented is very broad and involves many people in a management activities oriented organization (ranging from directors' boards to linear managers). It is a subset of topics that is to be read and learned by a majority of users' categories.
c. The core of the methodology integrates words quite unusual and new for the majority of Russian managers (Corporate / Enterprise Architecture, Business Engineering). In addition management research and practice have no conventional terms and concepts. Thus words and phrases used in the methodology and especially in the topic headings can be misunderstood and users will be unable to set a relation between their mental models and topics of the methodology.

In order to effectively communicate methodology with respect to the specialties described above Knowledge Navigator (KN) was created.

5 CASE STUDY: KNOWLEDGE NAVIGATOR FRAMEWORK AND DESCRIPTION

Input data for KN are content ontology and the very content.
In order to satisfy the requirements for effective knowledge representation KN – end-user solution – integrates three tools (Figure 1):

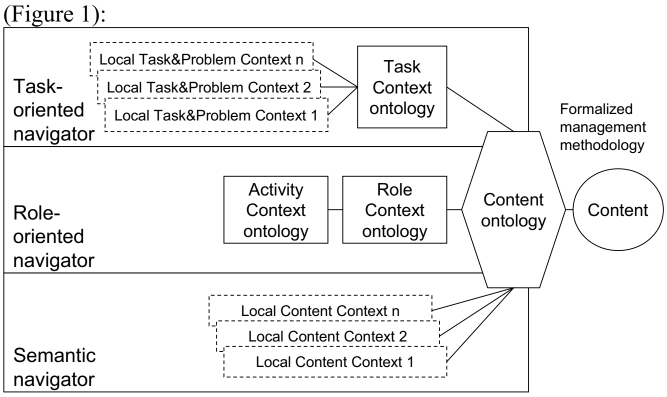

Figure 1. Knowledge Navigator Framework

1. Task-oriented navigator ("What for" – navigator)
   It helps users to choose topics to solve certain tasks and problems of organization.
   This navigator maps content ontology with Pragmatic context, which is represented in the form of Task Context ontology. But although the latter ontology results from the analysis made by a consulting company and is shared by the authors, it is not shared by prospective users and consequently does not satisfy Requirement 1 from [Section 3]. In order to help the users identify their local problems every node in Task Context ontology is mapped with a set of descriptive local task and problem contexts of users. These local contexts are given even

in user linguistics. Finally users of this navigator do two consecutive mappings, see Step 1 and Step 2 in Figure 2.

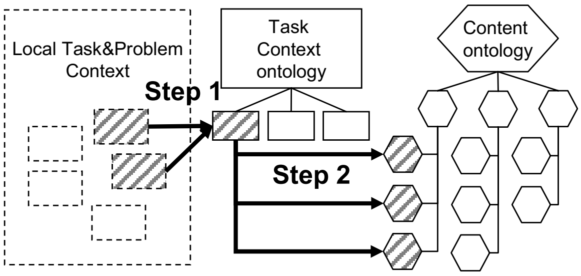

Figure 2. Task-oriented navigator - two consecutive mappings

Real-life example for shaded blocks from Figure 2 is represented in Table 1.

Table 1. Task-oriented navigator – example

Local Task&Problem Context	Task Context ontology	Impor-tance	Content Ontology /Topics
1. You might have encountered situations of complete chaos resulted from disorganization in your company. These cause the same problems to reoccur. -- 2. The strategy issues are left unheeded in your company. The main question your company managers are faced with is “how to cater to the clients’ order”	To establish order	○	Business Engineering and modeling
		△	Corporate Architecture as a control object
		⊙	Tools of Business Engineering

Importance:	⊙	Critical	○	Important	△	Useful

2. Role-oriented navigator (“Who” – navigator)
   It helps users to choose topics for learning with respect to their Roles in the organization.
   This navigator maps content ontology with Pragmatic context, which is represented in the form of Role Context ontology. Similarly to task-oriented navigator, Role Context ontology is ambiguous and polysemantic for the users, because Roles (nodes of Role Context ontology) can bear different responsibilities in different organizations. Thus the Role Context ontology is mapped with the elements derived from the next Pragmatic context - Activity Context ontology. The Activity Context ontology can be considered as shared by potential users, because all the management activities presented are typical for different organizations. Finally users of this navigator also do two consecutive mappings, see Step 1 and Step 2 in Figure 3.

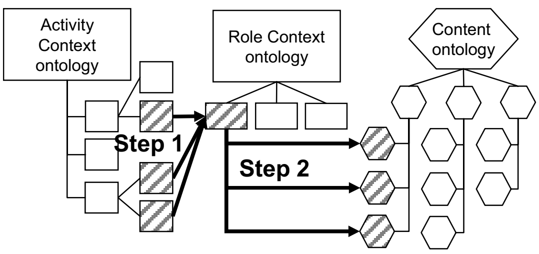

Figure 3. Role-oriented navigator - two consecutive mappings

Real-life example for shaded blocks from Figure 2 is represented in Table 2.

Table 2. Role-oriented navigator – example

Activity Context ontology	Role Context ontology	Impor-tance	Content Ontology /Topics
Perform external and internal analysis	Director of Business Development	⊙	Ideology of modern organization
Develop business strategy		○	Business Engineering and modeling
Develop and set organizational goals		△	Corporate Architecture as a control object

3. Semantic navigator (“What about” – navigator)
   This navigator helps users to relate topics in authors language with their knowledge and thus refine a subset of topics to learn. This navigator maps the Content ontology with the Local Content Contexts, which are represented by the keywords.

Namely this combination of 3 tools together with internal mapping will provide effective communication. Such a framework of KN takes into account knowledge communication specialties (problem) from [Section 4] and satisfies the requirements from [Section 3].

6 CONCLUSIONS

This paper suggested the requirements for effective knowledge representation based on mapping ontologies and context with respect to two types of the latter. It described a solution for real-life knowledge communication task called Knowledge Navigator. This solution illustrated consecutive mapping ontologies and contexts – mapping which was necessary to effectively communicate knowledge to different users, which solve different tasks and have different understanding of domain and background.

7 REFERENCES

[1] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, M. Sintek, Toward a Technology for Organizational Memories IEEE Intelligent Systems. – 1998. - №3, 40-48.
[2] A. Abecker, D. Apostolou, W. Maas, G. Mentzas, C. Reuschling, S. Tabor, Towards an Information Ontology for Knowledge Asset Trading Presented at the ICE 2003 - 9th International Conference of Concurrent Enterprising, Espoo, Finland, 16-18 June 2003
[3] P. Bouquet, F. Giunchiglia, F. Harmelen, L. Serafini, H. Stuckenschmidt, Contextualizing Ontologies, Journal of Web Semantics, 2004, Vol.1, №4.
[4] M. Eppler, Making Knowledge Visible Through Intranet Knowledge Maps: Concepts, Elements, Cases Proceedings of the 34th Hawaii International Conference on System Sciences - 2001
[5] F. Harmelen, Ontology Mapping: A Way Out of the Medical Tower of Babel? AIME 2005, pp. 1–4, 2005.
[6] T. Gruber, A translation approach to portable ontology specifications. Knowledge Acquisition, 1993, Vol. 5, 199- 220.
[7] C. Ghidini, F. Giunchiglia, Local models semantics, or contextual reasoning = locality + compatibility, Artif. Intell. 127 2 (2001) 221–259.
[8] F. Giunchiglia, P. Shvaiko, Semantic Matching. In The Knowledge Engineering Review Journal, vol. 18(3), pp. 265-280, 2003.
[9] M. Klein, *Combining and relating ontologies: an analysis of problems and solutions*, Workshop on Ontologies and Information Sharing, IJCAI'01, 2001, №4-5.
[10] C. Morris "Foundations of the Theory of Signs." International Encyclopedia of Unified Science, ed. Otto Neurath, vol. 1 no. 2. (Chicago: University of Chicago Press, 1938. Rpt, Chicago: University of Chicago Press, 1970-71).
[11] S. Staab, A. Maedche Knowledge Portals: Ontologies at Work. AI Magazine 2001, Vol. 22, №2, p. 63-75.
[13] M. Uschold, M. King, S. Moralee and Y. Zorgios The Enterprise Ontology AIAI, The University of Edinburgh, 1997.
[14] W. Vestal Knowledge Mapping: The Essentials for Success APQC: Publications. 2005.

Engineering a Brokering Framework for Providing Semantic Services to Agents on Lightweight Devices

Nikolaos I. Spanoudakis[1] and Pavlos Moraitis[2]

Abstract. This paper describes an approach towards allowing lightweight nomad devices like mobile phones to access semantic services that have either been advertised by agents or follow the semantic web services paradigm. The limitations of lightweight devices like lack of capability to process XML documents or to deal with complex data types and perform computationally demanding tasks are overcome by using this approach. Thus, we consider that when a user or agent is in a nomad or mobile context this approach can aid him in searching for and acquiring simple or complex - added value services from the web.

1 INTRODUCTION

There is a growing interest on the launching of agents on lightweight devices and that comes from many different business and research sectors, including the Ambient Intelligence, the infomobility, the collaborative working environments and others.

Lightweight devices pose certain limitations on the available resources (CPU speed, memory capacity, storage capacity, etc) for programs. Services are becoming semantic so that agents can adequately locate and execute them in order to achieve their goals. Semantic services imply the use of XML, RDF and OWL [13] technologies. The use of such technologies requires more than what is available on a lightweight device.

Brokering is the solution to this problem, since the broker can always be on a server computer side having access to needed computational and storage resources. The nomad devices residing agents need access to services that require computational power (for example filtering 100 hotels in order to present to a user the best 10). Such services are proposed to be offered by server-based provider agents. Important works from the agent technology domain, but also from that of the semantic web, have addressed the issue of brokering and matchmaking ([3], [11], [7]). However, these works lack the support for agents on the specific context of being resident on nomad, lightweight devices.

Our work builds on this previous work and provides a framework for defining services using the OWL-S [12] paradigm and making them available to lightweight devices. We use the FIPA-ACL [2] standard for defining the agent messages that are used by our novel interaction protocol. The content of the messages is encoded using the FIPA-SL [2] language for lightweight agents.

In section 2 we describe our approach in detail and we conclude with a discussion in section 3. We use italics in order to type concepts of the ontology that we developed, their properties and ACL message performatives.

[1] Singular Software SA, Greece, email: nspan@singular.gr
[2] René Descartes University, Paris, France, email: pavlos@math-info.univ-paris5.fr

2 THE BROKERING FRAMEWORK

In order to address our requirements we used the broker agent type [6], who can actively interface between the requester and provider agents by facilitating the requested service transaction. Thus, all communication between requester and provider agents has to go through the broker. In this process, the requester's identity is unknown to the service provider. Thus, assuming the business role of a service aggregator the broker services his customers using providers as resources. The service requesters are assumed to be aware of the services that they need. In our system, the role of the broker is to either select the best service for the requester, or to redirect the request to the appropriate broker. The added value of our approach is the service protocol that allows for anonymous brokering for agents in a nomad context adding, for the first time, the possibility to broker subscription services.

The matchmaking process is a subset of LARKS [11], suited for the nomad device applications domain. In this context, the requester is assumed to use the same ontology with the broker. Our process's novelty lies in the manipulation of the inter-agent messages content that is delivered using the LEAP protocol, which poses specific limitations and is in byte code format. Heterogeneous services are wrapped by service provider agents who advertise and offer services using the application domain ontology.

Moreover, brokers can be distributed and each one can specialize to a specific domain of services. We follow the notion of broker specialization of Infosleuth [9]. We model this requirement using a broker capability property concept allowing a broker to define its specialization in terms of service parameters constraints and share it with other brokers. The advantages of our approach compared to Infosleuth are a) brokers do not simply exchange their advertisements but define their special capabilities over the provided services in the domain, b) the requester agent doesn't have to define a search policy for the broker, and, c) compatibility with FIPA standards.

The way to profile the services, the matchmaking process and the brokering protocol are described in detail in the following paragraphs.

2.1 Service Profiling

For service profiling we follow the semantic web trend and thus are compatible to OWL-S. The service profile (SP) defines the type of the service (e.g. mapping service), describes input and output parameters, as well as preconditions and post-conditions. Here we would like to note that in the service parameters definition we have defined semantics for declaring a parameter as optional or mandatory.

2.2 The matchmaking process

Having defined the input requests and profiles we can proceed to defining the matchmaking process. We need to match a service advertisement to a service request. Two types of matching best serve our needs ([7], [11]): a) *the exact matching*, which demands that the advertised service has the same semantics and equal input/output parameters with the requested service, and b) the *plug-in matching*, which allows for the advertisement to have more input/*output parameters* than the ones requested. The exact matching is obviously always preferable.

Our matchmaking algorithm gradually filters the repository of advertisements until the one best to serve the request is found. Three types of filters, originally proposed by [11], are used: a) *Semantic Match (SM)* searches the service *profile* advertisements (PAs) for a service that matches the request (RP), b) *Profile Match (PM)* searches the PAs provided by the SM for input and output parameters that match those of the request. PM determines which PAs are exact or plug-in matches and sorts them accordingly, and, c) *Constraint Match (CM)* determines which of the PAs provided by the SM, match the constraints of a request. CM is performed to the sorted list provided by PM and either the first or all PAs that successfully match the constraints are selected depending on the broker's policy.

A special CM is needed before SM (named Pre-CM) so that the broker agent (BA) can determine if he can serve the request or it needs to redirect the request to another broker. Thus, broker capabilities are described as constraints for parameter values. For querying the PAs repository we use the RDQL (RDF Query Language) of the Jena open source tool [5].

Thus, according to our matchmaking algorithm, the broker first applies the pre-CM filter. If he can handle the request, he then sequentially applies the three other filters (SM, PM, CM) to his PA repository.

Technical challenges were related to this matchmaking process. The first was relevant to the transformation of a LEAP message to RDF format for filtering. In order to overcome it we use the LEAP codec for decoding a message at the broker side and then encode it again with the use of the RDF codec [4] (see an example in Figure 1). Thus, the request gains the necessary semantics so as to be processable by Jena. From that point forward the matchmaking process takes place and whenever the response of the service provider is ready, it is encoded at the broker side with the LEAP codec and sent to the lightweight agent requester.

Another problem that we had to overcome is that FIPA ACL allows for a single ontology to be included in the content of an ACL message. Thus, it is not possible to use different existing ontologies when defining the ACL protocols (e.g. import all OWL-S namespaces and use their concepts). That is why we added in our ontology all the concepts that we need in order to define a service profile similar to OWL-S. However, these are reusable since the Protégé tool [10] that we used in order to define our ontology doesn't associate namespaces to ontologies before deployment.

For describing an input/output parameter within a request for a service we created the *CallParameter* concept. There, we encountered another technical challenge related to the fact that we used the LEAP codec and thus, we could not add dynamically a value concept to a parameter property as we could easily do in an RDF document. This happens because the agent on the nomad device uses ontology java beans [1] in order to represent ontology concepts. These beans are normal Java classes containing properties that cannot be ambiguous, i.e. defined of type *Object* because the LEAP encoding and decoding process needs specific data types to instantiate as properties of concepts.


```
<?xml version="1.0"?>
<rdf:RDF   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
xmlns:fipa-rdf="http://www.fipa.org/schemas/FIPA-RDF#"
xmlns:O="http://imagine-it.eu/ontology#">
 <rdf:object>
  <fipa-rdf:CONTENT_ELEMENT>
   <rdf:Description>
    <rdf:type>http://imagine-
it.eu/ontology#RequestForEService</rdf:type>
    <O:agent>
     <rdf:Description>
      <O:name>broker@nspan2kp:1099/JADE</O:name>
     </rdf:Description>
    </O:agent>
    <O:requestEService>
     <rdf:Description>
      <O:serviceName>http://imagine-it.eu/ontology#
createMap</O:serviceName>
      <O:hasParameterIn>
       <rdf:Seq>
        <rdf:li>
         <rdf:object>
          <rdf:type>http://imagine-it.eu/ontology#
CallParameter</rdf:type>
          <O:withName>http://imagine-
it.eu/ontology#forCountry</O:withName>
          <O:withType>http://www.w3.org/2001/XMLSchema#
string</O:withType>
          <O:withStringValue>DE</O:withStringValue>
         </rdf:object>
        </rdf:li>
        <rdf:li>
         <rdf:object>
          <rdf:type>http://imagine-it.eu/ontology#
CallParameter</rdf:type>
          <O:withName>http://imagine-it.eu/ontology#
screenSize</O:withName>
          <O:withType>http://imagine-it.eu/ontology#
ScreenSize</O:withType>
          <O:withScreenSizeValue>
           <rdf:Description>
            <O:hasPixelsHeight>200</O:hasPixelsHeight>
            <O:hasPixelsWidth>320</O:hasPixelsWidth>
           </rdf:Description>
          </O:withScreenSizeValue>
         </rdf:object>
        </rdf:li>
       </rdf:Seq>
      </O:hasParameterIn>
     </rdf:Description>
    </O:requestEService>
   </rdf:Description>
  </fipa-rdf:CONTENT_ELEMENT>
 </rdf:object>
</rdf:RDF>
```


Figure 1. A Service Request Message (RDF)

We overcame this issue by defining all possible values that a parameter can have as properties of the *CallParameter* concept. The basic properties of the *CallParameter* in an OWL/RDF setting would be the *withName*, *withType* and *withValue*. In this case, however, we must cater for all possible types defined in our ontology concepts. Figure 1 shows an instance of a request message related to a specific application [8], which is based on the *http://imagine-it.eu/ontology#* namespace. The reader can observe the *hasParameterIn* RDF sequence element that is composed of a list of *CallParameter* elements that have a name (parameter *withName*), a type (it can be a simple data type such as string or a complex type like for example the *ScreenSize* type) and a value corresponding to the type.

For example, for the *ScreenSize* type (these types are also related to application [8]) the relevant property of *CallParameter* that is used is the *withScreenSizeValue*. Similarly the *forCountry CallParameter* is of type *string* and has the *withStringValue* property. Thus, the *CallParameter* concept has as many such properties as the number of the data type concepts defined in our ontology. However, for each instance the requester defines the *withName* and *withType* (*withType=PropertyType*) properties and the relevant *withPropertyTypeValue* property. It is obvious that a designer can define appropriate parameter types related to his own application.

2.3 The Service Protocol

The service provisioning protocol is presented in Figure 2 in the form of a FIPA interaction diagram [2].

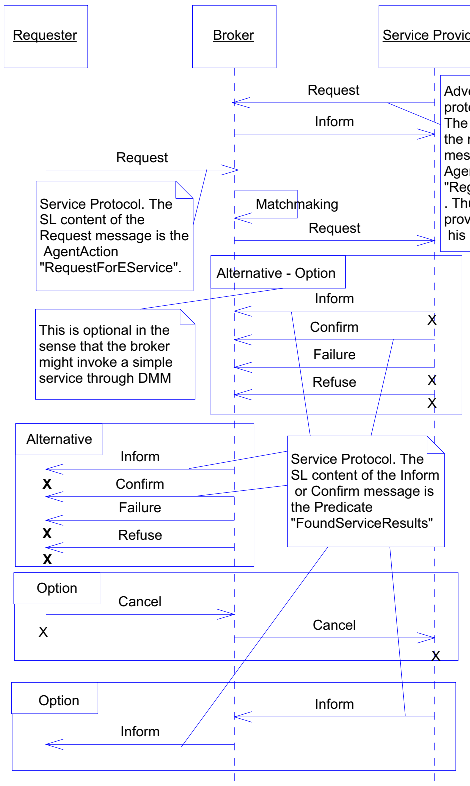

Figure 2. Service Protocol Definition

The service protocol can also be used for subscription services provisioning. We use the FIPA *Request, Inform, Refuse, Failure* and *Confirm* performatives. The important *AgentAction* and *Predicate* concepts [1] that are used as content in the ACL message are also presented. The participants are the Requester agent type (RE), the Broker agent type (BR) and the service provider agent type (SP). In the case of distributed brokering the SP is another broker, who considers the broker, who received the original request, as a RE (implementing the relevant part of the same protocol).

Finally, for the service subscription protocol, the broker always retains the recent addresses of the communicating agents so as to be able to forward new messages to the latest address of the requester agent. This is important since the requester agent is on a nomad device and usually is assigned a dynamic IP whenever he accesses the network.

3 DISCUSSION

We used this brokering framework in the context of IST project Im@gine IT in the infomobility sector domain [8]. An Im@gine IT prototype has been developed and deployed. Two added value service providers were developed.

This work is meant as an extension of important works in the brokering domain ([3], [11], [7]), towards offering semantic services to nomad devices. We provide a complete solution for the nomad devices service provisioning including not only simple services but also the delegation of complex tasks and subscription services. The solution is composed of a protocol, a service profiling scheme and the relevant matchmaking process.

As future work we aim to enrich the broker with the capability to use directly OWL-S services advertisements (along with those received by other agents) where the broker performs the service grounding himself.

Acknowledgements

We gratefully acknowledge the European Commission Information Society Technologies (IST) Programme and specifically the Integrated Project (IP) “Ambient Intelligence System of Agents for Knowledge-based and Integrated Services for Mobility Impaired users” (ASK-IT, IST-2003-511298) project for contributing in the funding of our work.

REFERENCES

[1] Caire, G., Van Aart, C., Bergenti, F., Pels, R. : Creating and Using Ontologies in Agent Communication. Workshop on Ontologies and Agent Systems at AAMAS 2002

[2] FIPA, Foundation for Intelligent Physical Agents, www.fipa.org

[3] Gomez, M., Abasolo, C., Plaza, E.: Domain-independent ontologies for cooperative information agents. In: Cooperative Information Agents V. LNAI 2128, Springer-Verlag, 2001, p. 118-129

[4] JADE, Java Agent Development Environment, http://jade.tilab.com/

[5] Jena, A Semantic Web Framework for Java, http://jena.sourceforge.net/

[6] Klucsh M., Sycara K.: Brokering and Matchmaking for Coordination of Agent Societies: A Survey. In Omicini et al (editor), Coordination of Internet Agents, Springer, 2001

[7] Li. L. and Horrocks, I.: A software framework for matchmaking based on semantic web technology. Int. Journal of Electronic Commerce, Vol. 8, No 4, 2004

[8] Moraitis, P., Petraki, E. and Spanoudakis, N.: An Agent-Based System for Infomobility Services. In: 3rd European Workshop on Multi-Agent Systems (EUMAS2005), Brussels, Belgium, December 7 - 8, 2005

[9] Nodine, M., Bohrer, W., Hee Hiong Ngu, A.: Semantic Brokering over Dynamic Heterogeneous Data Sources in InfoSleuth. In Proc. of the International Conference on Data Engineering, Sydney, Australia, March 23-26, 1999

[10] Protégé, An Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu

[11] Sycara, K., Widoff, S., Klusch, M. and Lu, J.: LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. JAAMAS, 5, 173–203, 2002.

[12] The OWL Services Coalition (Martin, D. et al.): Bringing Semantics to Web Services: The OWL-S Approach, Proc. of the 1st Int. Workshop on Semantic Web Services and Web Process Composition (SWSWPC2004), July 6-9, 2004, San Diego, California, USA.

[13] W3C, The World Wide Web Consortium, www.w3c.org

On the Modeling of Context-Rules with WSML

Reto Krummenacher and **Holger Lausen** and **Thomas Strang** [1]

Abstract. Modeling context information based on formal descriptions is a core aspect of service integration and interoperability, in particular in pervasive computing environments. In this paper we present an improved and simplified version of the Context Ontology Language modeled in WSML-Rule to show the potential use of context rules in pervasive computing applications and in particular as part of Semantic Web service descriptions.

1 INTRODUCTION

The trend towards pervasive computing is driving a need for services and service architectures that are aware of the context of the different actors (users, service providers, or third parties and their environments) involved in a service interaction: vicinity, location, QoS, ownership, time. For instance, context information can be used to reduce the amount of required user or service-service interactions, as well as to improve the user experience. A key accessor to context information in any context-aware system is a well designed model to describe contextual facts and contextual interrelationships. The context modeling approach applied in this paper is derived from the Context Ontology Language (CoOL, [7]). CoOL is based on the Aspect-Scale-Context (ASC) model also introduced in [7]. ASC defines a very simple context model in form of an extendable umbrella vocabulary that is shown to increase interoperability on the contextual level.

In this paper we improve and simplify the context modeling language by evolving its definitions based on recent advances in the field of Web-rule languages. We look in particular at the application of CoOL in combination with rule-based WSML variants [1]. This allows us to update the well designed context model and to bind it to a language family that is part of a large framework of Semantic Web languages. The WSML family of languages is a member submission to the W3C, and although it does not have the status of an official standards recommendation, we expect to be able to easily map our results into the ongoing work of the Rule Interchange Format working group [3], which will eventually endorse an official standard. Furthermore the application of rule languages allows for a simplified Context Ontology Language through the use of meta-modeling, where a concept itself can have attribute values just like any particular instance (cf. Section 4).

The paper is organized as follows. In Section 2 a short introduction to WSML is given. Section 3 provides more information about the Aspect-Scale-Conext model and the derived Context Ontology Language (CoOL). In Section 4 we show how CoOL can be modeled using WSML-Rule and how to define context-rules. We also look at possible application areas of context-rules, in particular in the area of Semantic Web services, where the WSML family resulted from. Finally we conclude with Section 5 and provide a short outlook at where and how the ideas of this paper will be further explored.

2 WSML-RULE LANGUAGE

The Web Service Modeling Language WSML [1] provides a framework for the modeling of ontologies and Semantic Web services based on the conceptual model of the Web Service Modeling Ontology WSMO [5]. WSML defines two rule-based language variants that are of interest to the issues of this paper. The first rule-based variant, WSML-Flight, semantically corresponds to the Datalog fragment of F-Logic, extended with inequality in the body and locally stratified negation under the perfect model semantics [6]. The second, WSML-Rule, extends WSML-Flight to the logic programming subset of F-Logic which allows the use of function symbols and unsafe rules (i.e., there may be variables in the rule head which do not occur in the body).

A WSML rule has the common form of *head :- body*. We illustrate this with the following example which states that every woman (rule body) is a human being (rule head):

?x **memberOf** Human :− ?x **memberOf** Woman.

Further technical details about the language are available in [1].

3 CoOL: CONTEXT ONTOLOGY LANGUAGE

The context description language applied in this paper was described in [7] and is based on the Aspect-Scale-Context model introduced in the same dissertation. On an generic level an aspect is a dimension of the situation space that is used as a collective term for information objects having the same semantic type. A scale is then seen as an unordered set of objects defining the range of valid context information instances. In other words, a valid context information with respect to an aspect is one of the elements of the aspect's scales. This results in a number of aspects that aggregate one or more scales, where each scale aggregates one or more pieces of context information. The three concepts that constitute the CoOL-core ontology are interrelated by use of the attributes hasAspect, hasScale, hasMember and usedByScale (cf. Figure 1).

Through the combination of meta-data instances, CoOL allows the provision of higher order context information or the binding of quality measures. In [7] meanError, timestamp and hasQuality were proposed for any context information instance.

A particular strength of the presented context model not yet mentioned is the infrastructure defined to map semantically related scales of one aspect or to combine and interlink different scales to new scales of hybrid aspects. There are two types of operations in CoOL: (1) IntraOperations that provide translations from one scale to another, e.g. from Kilometer to Miles of a DistanceAspect, and (2)

[1] Digital Enterprise Research Institute, University of Innsbruck, Austria, email: firstname.lastname@deri.org

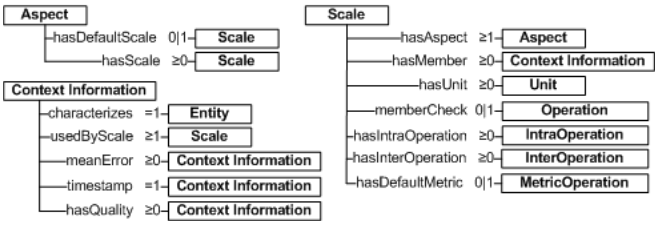

Figure 1. The ASC ontology from [7]

InterOperations that allow for example the definition of a KilometerPerHourScale of the SpeedAspect as a combination of a KilometerScale and an HourScale.

More details about the ASC model are provided in the next section where we first of all discuss some simplifications and improvements.

4 CONTEXT RULE MODELLING IN WSML

In this section we present CoOL written in WSML-Rule[2] (Listing 1). Note first of all a minor change in the model with respect to the original ontology (Figure 1): we feel that a context information is not used by a scale, but rather that the context information is encoded as given by a scale. Hence, we suggest to use the attribute inScale instead.

Listing 1. CoOL-core written in WSML-Flight.

```
concept Aspect
  hasDefaultScale ofType (0 1) Scale
  hasScale ofType Scale

axiom DefaultScaleSubScale definedBy
  ?a[hasScale hasValue ?s]
    :− ?a[hasDefaultScale hasValue ?s] memberOf Aspect.

concept Scale
  hasAspect inverseOf(hasScale) ofType (1 ∗) Aspect
  hasMember ofType ContextInformation
  hasUnit ofType Unit
  memberCheck ofType _iri
  hasIntraOperation _iri
  hasInterOperation ofType _iri
  hasDefaultMetric ofType (0 1) _iri

concept ContextInformation
  characterizes impliesType (1) Entity
  inScale inverseOf(hasMember) ofType (1 ∗) Scale
  meanError ofType ContextInformation
  timestamp ofType (1) ContextInformation
  hasQuality ofType ContextInformation
```

Based on the core concepts of CoOL it is now possible to define particular aspects, scales and pieces of context information. Listing 2 shows the necessary concepts and instances to gather information about distances in either kilometer or miles. Using the WSML language constructs like **inverseOf** keeps the definition of a domain ontology for distance measurements short and simple without loosing e.g., the aspect-scale or scale-aspect bindings. The context information containers are explicitly given by the axioms that bind them to a given distance scale (Listing 2).

CoOL has so-called *memberCheck* operations (Figure 1) that ensure correctly scaled values for context information (i.e. that they obey the type of the scale). In WSML such constraints can in simple cases directly be expressed within the conceptual syntax. In our example the values are constrained to the datatype float directly in the axiomatic definition of *KmCI* or *MilesCI*. The semantics of WSML ensures that if instances exist in a model that do not obey these constraints, such a model is inconsistent and in fact no valid model at all. For more complex value constraints it is always possible to bind an external operation to the model, as will be described later in this section.

Listing 2. An example of CoOL-WSML for distance information

```
instance DistanceAspect memberOf Aspect
  hasDefaultScale hasValue KilometerScale

instance KilometerScale memberOf Scale
  hasAspect hasValue DistanceAspect

instance MilesScale memberOf Scale
  hasAspect hasValue DistanceAspect

axiom defaultScaleKmCI definedBy
  ?kci[inScale hasValue KilometerScale, value ofType _float ]
    :− ?kci memberOf KmCI .

axiom defaultScaleMiCI definedBy
  ?mci[inScale hasValue MilesScale, value ofType _float ]
    :− ?mci memberOf MilesCI .
```

For a better understanding we first elaborate on the example in Listing 2. There is one aspect, the DistanceAspect, defined in the domain ontology that represents one possible context dimension: spatial distance. The default scale for distance measurements is defined to be the KilometerScale. A second possible scale would be the MilesScale. The aspects and scales are modeled as instances of the CoOL-core Aspect respectively Scale concepts, while the context information objects are implicitly defined as concepts (KmCI, respectively MilesCI) to provide containers for all collected instances, i.e. pieces of information.

As shown in Listing 3, it is possible to directly axiomatize simple intra operations within WSML, they can be modeled by rules, which transparently make values of context information available in different scales (e.g. Miles and Kilometer). The axiom *km2miOperation* infers for example the context information in the MilesScale from some in the KilometerScale. We use a function symbol to generate an identifier for inferred context information to distinguish between inferred and measured information. The rule states that every measurement that is taken using the KilometerScale is equivalent to a value in the MilesScale divided by 1.609.

Listing 3. Axiomatic IntraOperations.

```
axiom km2miOperation definedBy
  km2mi(?info)[inScale hasValue MilesScale, value hasValue ?mi,
      characterizes hasValue ?entity ] memberOf MilesCI
    :− ?info[inScale hasValue KilometerScale, value hasValue ?km,
      characterizes hasValue ?entity ] memberOf KmCI
    and wsml#numericDivide(?mi,?km,1.609) and naf ?info = mi2km(?i) .
```

In order to provide the same information for intra-scale operations as in [7] we suggest to use non-functional properties to annotate the mapping axioms. The *fromScale* property indicates the source scale, while *toScale* provides a link to the target scale. Most intra-operations demand a simple value transformation to cope with different units. Similar mappings exist for a TemperatureAspect where Kelvin, °C and °F would have to be interlinked to make the information compatible even though it results from heterogeneous data sources.

Before looking at the definition of context rules we shortly add some distance measurements to our knowledge base. The distance is either given by an explicit measured instance or by an inferred instance generated on-the-fly by an appropriate axiom:

```
axiom measurements definedBy
  _# memberOf KmCI[value hasValue ?d, characterizes hasValue DistAB]
    and distKM(?d,A,B) .
  _# memberOf KmCI[value hasValue 14, characterizes hasValue DistAC] .
  _# memberOf MilesCI[value hasValue 8.5,
      characterizes hasValue DistAD] .
```

[2] The complete listings are at http://members.deri.org/~retok/cool/

By now the reader should be familiar with the context modeling ontology and with the way context domains and context information are defined using WSML-Rule.

A context rule is an axiom that is defined by an implication where the body is a set of conditions using the context information in the knowledge base. Rules either infer new knowledge or return information if posted in form of queries. The following example queries distance entities that represent nearby locations. The resulting distance value shall be provided by an MilesCI instance of a scale that belongs to the aspect DistanceAspect and shall be smaller than 10 miles:

```
? − ?_c[ characterizes  hasValue ?entity ,  value  hasValue ?distMiles ,
          hasScale  hasValue ?s]  memberOf MilesCI
     and ?_s[hasAspect hasValue DistanceAspect] and ?distMiles  < 10 .
```

The query returns for the given measurements the following matches[3]:

?entity	?distMiles
DistA2C	8.7
DistA2D	8.5

In WSMO [5] the vocabularies, constraints and logical expressions that are defined in ontologies are used to describe the functionality (capability) and interfaces of Web services. The just defined query could be used to include restrictions on the spatial distance between the service provider and requester. It could for example be envisioned that a pizza delivery service only accepts orders from clients that call from at most ten miles from the pizza store. Hence, a precondition of such a pizza ordering service would include a constraint that uses the context rule to ensure the desired maximal spatial distances.

This leads us to another interesting feature that the WSML framework provides. WSMO and in consequence WSML were developed to annotate Web service descriptions. In [7] the various operations are offered by external services that are linked into the model by use of operation bindings (Figure 1). We have already shown that many IntraOperations and member checks can be modeled by axiomatic expressions, while for the more complex ones, as well as for InterOperations and MetricOperations WSML provides us with the means of Web service descriptions within the same framework and thus based on the same notations and vocabularies.

Listing 4. A Web service description to link InterOperations

```
webService _"http :// www.example.org/interOpService"

  capability
    precondition definedBy
        ?i1 memberOf KilometerScale and ?i2 memberOf HourScale .
    postcondition definedBy
        ?o memberOf KmPerHourScale .
```

The shown service description (Listing 4) contains a capability description that uniquely indicates the constraints on the input and output parameters of the service that computes the kilometer per hour scale (Listing 5). The description of the grounding and interaction patterns with the Web service are assumed to be given in an external file, as this would exceed the scope of this paper. The goal is to reconsider the strength of CoOL and to show the advantages of modeling it with WSML, in particular with WSML-Rule.

Listing 5. A scale definition with IntraOperation binding

```
instance KilometerPerHourScale memberOf Scale
  hasAspect hasValue SpeedScale
  hasInterOperation hasValue _"http :// www.example.org/interOpService"
```

[3] WSML-Rule reasoner: http://tools.deri.org/wsml/rule-reasoner

5 CONCLUSION

Context-awareness, and as a crucial intermediate step the provision of concise context models, is a core research area of pervasive computing. Encoding context information by use of ontologies allows for formal descriptions of characteristics and states of entities. The ASC model and the derived Context Ontology Language CoOL provide a simple and extensible model based on aspect-scale-context interrelations.

In this paper we used the rule-based languages of the WSML language family to improve and simplify the language bindings proposed in [7]. The use of meta-modeling and the fact that WSML provides a set of languages that can be mapped to various types of logical formalisms which are already well integrated into the rule efforts of the Semantic Web allows for an even more concise, yet simultaneously extensible and globally applicable, umbrella framework for the modeling of context information.

This is exactly the convergence of technologies that is envisioned to be necessary to fully explore the use of context information in the field of service interoperability and information exchange on the context level. The generic character of the ASC model and the well-integratedness of WSML into the Semantic Web standardization activities allows this combined approach to become a context-modeling framework that could provide the backbone for large-scale context-aware applications on the Web. The requested and provided context information of various heterogenous information sources, sinks and services can hence be combined, processed and mapped on the machine level. In that sense, the ideas presented are expected to also improve Semantic Web services frameworks like WSMX [2] or service coordination infrastructures like Triple Spaces [4] by allowing their components to become context-aware. The upcoming work is thus concerned with enhancing the functionality-centered static descriptions of Web services to additionally consider dynamic characteristics like location, connectivity or quality to provide improved discovery, selection and coordination of services — a requirement for the access and composition of services in ubiquitous computing environments.

ACKNOWLEDGEMENTS

The work is funded by the European Commission under the projects DIP, Knowledge Web, SEKT, and ASG; by the Austrian Federal Ministry for Transport, Innovation, and Technology under the FIT-IT projects RW^2 and TSC.

REFERENCES

[1] J. de Bruijn, H. Lausen, A. Polleres, and D. Fensel: The web service modeling language: An overview, Proc. 3rd European Semantic Web Conf., June 2006.

[2] E. Cimpian, and M. Zaremba (eds.): Web Service Execution Environment (WSMX), W3C Member Submission, June 2005.

[3] S. Hawke (ed.): Rule Interchange Format Working Group Charter, W3C Semantic Web Activity, 2005.

[4] R. Krummenacher, M. Hepp, A. Polleres, Ch. Bussler, and Dieter Fensel: WWW or What is Wrong is with Web Services, Proc. 3rd European Conf. on Web Services, 2005.

[5] H. Lausen, A. Polleres, and D. Roman (eds.): Web Service Modeling Ontology (WSMO), W3C Member Submission, June 2005.

[6] T. C. Przymusinski. On the declarative and procedural semantics of logic programs. Journal of Automated Reasoning, 5(2), 1989.

[7] Th. Strang: Service interoperability in ubiquitous computing environments. PhD dissertation, LMU Munich, 2003.

Quality Extensions and Uncertainty Handling for Context Ontologies

Davy Preuveneers and **Yolande Berbers** [1]

Abstract. Context, by nature, involves real world entities and is therefore subject to uncertainty and inaccuracies. Ontologies are often used to model context in a formal way in order to achieve a shared semantic understanding of concepts and the relationships that hold among them. However, they lack support for representing ambiguous context and appropriate comparison algorithms. As such, context-aware applications may make the assumption that the context they use is completely accurate. In this paper we propose a simple and lightweight yet generic approach to extend context ontologies with quality of context properties and discuss the use of these quality properties for context ontology matching under uncertainty using fuzzy set theory. We illustrate the proposed extensions and uncertainty mechanisms with a small example where uncertain spatio-temporal coverage is combined with other contextual properties.

1 INTRODUCTION

Context-awareness has been drawing much attention from researchers in the ubiquitous and pervasive computing domain [12] as context has become a key ingredient to create a whole range of smart entertainment and business applications that are more supportive to the user. Context [4] has been defined as any information that can be used to characterize the situation of an entity. Humans take this context information into account rather intuitively, whereas context-aware applications require an explicit model to take advantage of context information for non-intrusive decision making and adaptation [9]. Imperfections in the context data can cause incorrect or unintended application behavior as relationships between similar context properties become uncertain. For example, the precision of a coordinate based positioning system is required to decide whether a given position matches with a location such as *'at the office'*.

In this paper we propose to extend context ontologies with quality of context properties and discuss a lightweight and generic approach for matching under uncertainty that is simple enough to be implemented and used on resource constrained devices, such as PDAs. The remainder of this paper is organized as follows. In section 2 we describe related work on quality of context and reasoning with uncertainty. Section 3 discusses how quality of context aspects are introduced into our context ontology. Section 4 describes the use of membership functions based on the concept of fuzzy set theory to achieve advanced matching mechanisms for context ontologies in the presence of uncertainty. In section 5 we conduct an experiment illustrating uncertain spatio-temporal coverage combined with other contextual properties to validate the matching mechanisms in more advanced scenarios. We conclude with section 6.

[1] Katholieke Universiteit Leuven, Belgium, email: {davy.preuveneers, yolande.berbers}@cs.kuleuven.be

2 RELATED WORK

In this section we focus on those contributions on quality of context and uncertainty management for mediation of ambiguous context that are most related to the work presented in this paper. This work is based on the ideas presented in Buchholz *et al.* [11], where the authors identify parameters that quantify the Quality of Context (QoC) and the inevitable uncertainty of sensed values for individual context properties:

- *Precision:* describes how sharply defined a measurement is stated and what the difference is with the actual value in the real world.
- *Probability of correctness:* estimates how often the context information is unintentionally wrong due to internal errors.
- *Trust-worthiness:* describes the reliability of the entity that may have persistently provided incorrect information in the past.
- *Resolution:* describes the granularity of the information and the inability to offer information with a finer detail.
- *Up-to-dateness:* describes the age of information which can be used to decide on the temporal relevancy in a particular situation.

Henricksen *et al.* [6] explore the problem of imperfect context information and characterize the following four types and sources of imperfect context information: *Unknown*, *Ambiguous*, *Imprecise* and *Erroneous*. The first two types of imperfection are new, whereas the latter two types combine several Quality of Context properties on the list of the work by Buchholz *et al.*.

In [7] Parsons describes qualitative methods for reasoning with various types of imperfect information and argues that qualitative methods have the advantage to not require precise numerical information, but instead to rely on abstractions such as interval values and information about how values change.

Chalmers *et al.* show in [1] how context can be formulated in the presence of uncertainty using interval arithmetic for numerical context values, and analogously using trees with abstract values for context ontologies. The authors define *within* and *overlap* relationships between actors and context objects both for numerical and abstract values in order to compare context information.

3 EXTENDING ONTOLOGIES WITH QUALITY PARAMETERS

Ontologies and the Web Ontology Language (OWL) are very popular for a systematic arrangement of context concepts and the relationships that hold among them [10, 2, 5]. In our previous work [3] we defined an OWL context model specifying *User*, *Platform*, *Service*, *Environment* and related concepts to provide a shared semantic understanding for context-driven adaptation of mobile services. Our context system [8] is able to gather and interpret this information. In

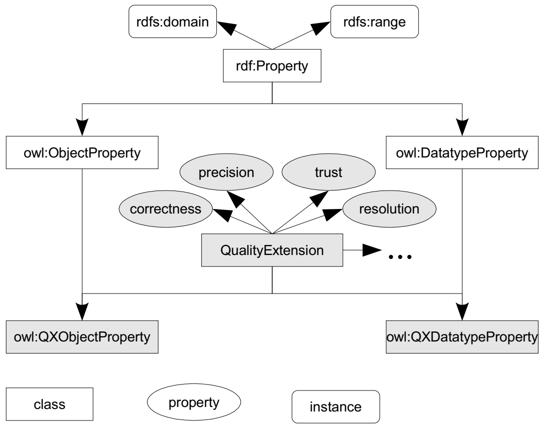

Figure 1. Extending the OWL language with QoC properties

the case of uncertainty in the gathered information, the context-aware system needs context quality parameters in OWL in order to determine a high confidence of correctness of matching context information. We will now show how the Quality of Context (QoC) parameters discussed in [11] are modeled by means of two new property types, *QXObjectProperty* and *QXDatatypeProperty*. Both property types inherit from the *DatatypeProperty* and *ObjectProperty* OWL language constructs, as well as from a self-defined class *QualityExtension* which models the Quality of Context parameters *precision*, *correctness*, *trust* and *resolution* as *DatatypeProperties*:

```
<owl:Class rdf:ID="QualityExtension" />

<owl:DatatypeProperty rdf:about="#precision">
  <rdfs:domain rdf:resource="#QualityExtension" />
  <rdfs:range rdf:resource="&xsd;#int" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#correctness">
  <rdfs:domain rdf:resource="#QualityExtension" />
  <rdfs:range rdf:resource="&xsd;#int" />
</owl:DatatypeProperty>
...

<owl:Class rdf:ID="QXDatatypeProperty">
  <rdfs:subClassOf rdf:resource="&owl;#DatatypeProperty" />
  <rdfs:subClassOf rdf:resource="&owl;#QualityExtension" />
</owl:Class>
<owl:Class rdf:ID="QXObjectProperty">
  <rdfs:subClassOf rdf:resource="&owl;#ObjectProperty" />
  <rdfs:subClassOf rdf:resource="&owl;#QualityExtension" />
</owl:Class>
```

See Figure 1 for an overview of the property inheritance hierarchy. The QoC parameters of e.g. a sensor that instantiates the temperature concept in our context ontology [3] are modeled as follows:

```
<owl:Class rdf:ID="Sensor" />

<qx:QXDatatypeProperty rdf:about="#hasTemperature">
  <rdfs:domain rdf:resource="#Sensor" />
  <rdfs:range rdf:resource="&xsd;#int" />
  <qx:precision>95</qx:precision>
  <qx:correctness>100</qx:correctness>
  <qx:resolution>1</qx:resolution>
  ...
</qx:QXDatatypeProperty>
```

4 MATCHING IN THE PRESENCE OF UNCERTAINTY WITH FUZZY SETS

In the real world context information can be vague, imprecise, uncertain, ambiguous, inexact, or probabilistic in nature. We therefore

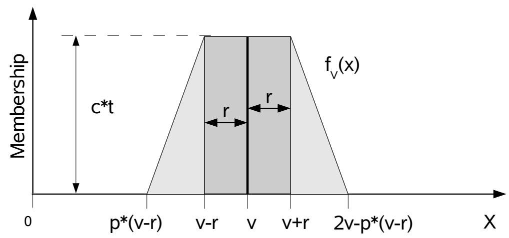

Figure 2. Membership function for a single sensed value with given Quality of Context parameters

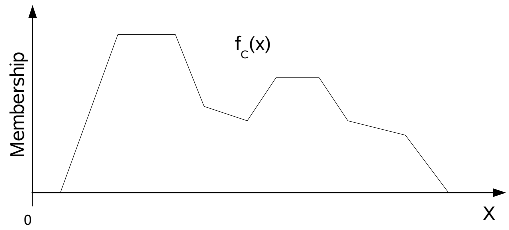

Figure 3. A fuzzy set C as a averaged sum of single fuzzy sets

need appropriate matching algorithms that take into account the imperfect nature of context when taking appropriate actions. In this section, we will show how we use concepts of fuzzy set theory of Zadeh [13] and define membership functions based on the quality of context parameters defined in the previous section.

4.1 Modeling a fuzzy context concept

In classical set theory the membership of an element to a set can be clearly described. In fuzzy set theory, an element belongs to a set with a certain possibility of membership. Age is a typical example of a fuzzy concept. There is no single quantitative value or clear boundary defined for the term *young*: age 25 can be young for some, while age 30 can be young for others. However, age 1 is definitely young, while age 100 is is definitely not young.

We can model the membership function for a single sensed value using the Quality of Context parameters in a similar way. Assume a sensed value v has a precision p, a probability of correctness c, a trust-worthiness t and a resolution r, with $0 \leq p, c, t \leq 1$, then we define the following symmetric membership function $f_V(x)$ with $x \in X$ for the sensed value v as in Figure 2. Note how the Quality of Context parameters change the crisp sensed value into an interval with a particular symmetric shape of the fuzzy set.

4.2 Aggregation and matching of fuzzy concepts

If a contextual concept C is defined by set of N measured values v_i then we can improve the accuracy of its membership function by using the aggregated membership of this concept $f_C(x)$ with $x \in X$ defined as the averaged sum of $f_{V_i}(x)$:

$$f_C(x) = \frac{\sum f_{V_i}(x)}{N} \qquad with \quad x \in X$$

For example, our WiFi location sensor uses multiple *Received Signal Strength Indication* (RSSI) values as a distance measurement to known access points and models them as fuzzy sets. An example of such an averaged sum of these fuzzy sets is shown in Figure 3. Note that the aggregated fuzzy set is no longer symmetric.

We define a match between two sensed values with fuzzy sets A and B and membership functions $f_A(x)$ and $f_B(x)$ based on the intersection of fuzzy sets A and B. The intersection [13] is a fuzzy set $C = A \bigcap B$ with a membership function $f_C(x) = f_A(x) \bigwedge f_B(x)$ which is defined as follows:

$$f_C(x) = f_A(x) \bigwedge f_B(x) = \mathrm{Min}[f_A(x), f_B(x)]$$

Two fuzzy concepts match if their overlapping area is larger than a user-defined and context-specific threshold α:

$$0 \leq \alpha \leq \frac{\int_X f_C(x)}{Min[\int_X f_A(x), \int_X f_B(x)]} \leq 1 \qquad with \quad x \in X$$

Of course, when one of the membership functions is $f(x) = 0$ or when the overlap is zero, then there is no need to calculate this ratio.

5 EVALUATION

This subsection discusses the scenario used for a preliminary evaluation of the uncertainty mechanisms for matching context information. A PDA enabled with WiFi networking is used for *Received Signal Strength Indication* (RSSI) based location-awareness. The computer science building has about 100 offices, labs and meeting rooms and is equipped with 7 access points for wireless Internet access on all 5 floors. In the first step we trained the system by walking around in the building and taking about 10 measurements for several offices.

We determined the Quality of Context parameters based on a long test run while remaining at the same location. We looked for outliers in the sampled data, calculated the mean and variation in the data and estimated the values of the QoC parameters as follows:

- Precision: 95%
- Probability of correctness: 90%
- Trust-worthiness: 100%
- Resolution: 3 dBm

Using this information, the average fuzzy set for each of the access points that were seen in a particular office was calculated. After ordering the overlap ratios by decreasing order, and selecting the fuzzy set with the highest overlapping ratio, the locations matched, although non of the new RSSI measurements was exactly equal to a previously encountered measurement at the same location.

In a second test scenario which illustrates spatio-temporal coverage, my PDA informs the instant messaging client on my desktop system on my whereabouts and adjusts my status accordingly. I usually have lunch around 12h30 and 13h00 together with my colleagues in a room which is also used for meetings. Both time and place should match in order for my client to change to the *'out for lunch'* status. If only the location matches, then my status should be *'in a meeting'*. Otherwise, if I am not in my office, I will *'be right back'*. Both location and time are modeled as fuzzy sets.

This simple test case with multiple fuzzy sets being matched worked fine in 4 out of 5 cases. On one day I had lunch at 13h30, but had a meeting before at the same place. The instant messaging client decided too early that I was out for lunch, and claimed that I had a meeting while I was still having lunch. This was due to the fact that the precision for the lunch time was set to high in order to match.

In the end, this simple approach using fuzzy set matching worked rather well for this particular application. However, for a large number of fuzzy sets that have to match at the same time, it becomes very difficult to decide which context information matches best as more and more scenarios will become equally likely.

6 CONCLUSION

In this paper we have proposed a simple and lightweight extension to the OWL language to model quality of context properties in order to deal with ambiguous and imperfect context information. We have discussed the use of these quality parameters in automated uncertainty reasoning to achieve more advanced matching mechanisms for context ontologies. This automated uncertainty reasoning was based on concepts of fuzzy set theory. We have illustrated the proposed ontology extensions and the fuzzy comparing algorithms with small examples which included spatio-temporal coverage as fuzzy sets.

The proposed matching mechanisms are still a work in progress, but worked as expected for the examples. Difficulties are assumed to arise when the number of fuzzy sets involved in a single contextual condition is going to increase. We therefore will further continue to refine the membership functions by including the likelihood of context information in order to reduce to possible scenarios that may match under particular circumstances. One improvement that may proof to be useful is the inclusion of likelihood of events. This will better differentiate the likelihood of fuzzy matches.

REFERENCES

[1] D. Chalmers, N. Dulay, and M. Sloman, 'Towards reasoning about context in the presence of uncertainty', in *Proceedings of Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004*, (2004).

[2] Harry Chen, Tim Finin, and Anupam Joshi, 'An ontology for context-aware pervasive computing environments', *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, (2003).

[3] D. Preuveneers, et al., 'Towards an extensible context ontology for ambient intelligence', in *Second European Symposium on Ambient Intelligence*, volume 3295 of *LNCS*, pp. 148 – 159, Eindhoven, The Netherlands, (Nov 8 – 11 2004). Springer.

[4] Anind K. Dey, 'Understanding and using context', *Personal Ubiquitous Comput.*, **5**(1), 4–7, (2001).

[5] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, California, USA, January 2004.

[6] Karen Henricksen and Jadwiga Indulska, 'Modelling and using imperfect context information', in *PERCOMW '04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, p. 33, Washington, DC, USA, (2004). IEEE Computer Society.

[7] Simon Parsons, *Qualitative Methods for Reasoning under Uncertainty*, The MIT Press, August 2001.

[8] Davy Preuveneers and Yolande Berbers, 'Adaptive context management using a component-based approach', in *Proceedings of 5th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS2005)*, Lecture Notes in Computer Science (LNCS). Springer Verlag, (June 2005).

[9] Davy Preuveneers and Yolande Berbers, 'Automated context-driven composition of pervasive services to alleviate non-functional concerns', *International Journal of Computing and Information Sciences*, **3**(2), 19–28, (August 2005).

[10] Thomas Strang, Claudia Linnhoff-Popien, and Korbinian Frank, 'CoOL: A Context Ontology Language to enable Contextual Interoperability', in *LNCS 2893: Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003)*, volume 2893 of *Lecture Notes in Computer Science (LNCS)*, pp. 236–247, Paris/France, (November 2003). Springer Verlag.

[11] T. Buchholz, A. Kupper and M. Schiffers, 'Quality of context: What it is and why we need it', in *Proceedings of the 10th Workshop of the OpenView University Association: OVUA'03*, Geneva, Switzerland, (July 2003).

[12] Mark Weiser, 'The Computer for the Twenty-First Century', *Scientific American*, 99–104, (September 1991).

[13] Lotfi A. Zadeh, 'Fuzzy sets', *Information and Control*, (8), 338–383, (1965).

Combining Ontology Alignment Metrics Using the Data Mining Techniques

Babak Bagheri Hariri and **Hassan Sayyadi** and **Hassan Abolhassani** and **Kyumars Sheykh Esmaili**[1]

Abstract. Several metrics have been proposed for recognition of relationships between elements of two Ontologies. Many of these methods select a number of such metrics and combine them to extract existing mappings. In this article, we present a method for selection of more effective metrics – based on data mining techniques. Furthermore, by having a set of metrics, we suggest a data-mining-like means for combining them into a better ontology alignment.

1 Introduction

Ontology Alignment is an essential tool in semantic web to overcome heterogeneity of data, which is an integral attribute of web. In [2], Ontology Alignment is defined as a set of correspondences between two or more ontologies. These correspondences are expressed as mappings, in which *Mapping* is a formal expression, that states the semantic relation between two entities belonging to different ontologies. There have been several proposals for drawing mappings in Ontology Alignment. Many of them define some metrics to measure *Similarity* or *Distance* of entities and find existing mappings using them [4]. To extract mappings, in most of these methods, couples having Compound Similarity higher than a predefined threshold – after applying a number of constraints – are selected as output. [4] contains a number of such methods.

In this paper, given several similarity metrics we are trying to determine which of them is best for a particular data set, using data mining techniques. In order to do that, we train our techniques on some mappings for which we have a *gold standard* alignment, determining which metric is the best predictor of the correct alignment. We consider such metrics to be the best, and calculate *Compound Similarity* using them.

The rest of this article is organized as follows. In section 2, a review of related works in evaluation of existing methods and calculation of compound similarity are given. Section 3 reports our proposed method. In section 4 an example of applying this method is shown. Finally in section 5, discusses on its advantages and disadvantages are explained.

[1] Semantic Web Laboratory, Computer Engineering Department, Sharif University of Technology, Tehran, Iran, email: {hariri,sayyadi}@ce.sharif.edu, abolhassani@sharif.edu, shesmail@ce.sharif.edu

2 Existing Works

Works on metric evaluation as well as a method for aggregating results of different metrics is introduced in this section.

2.1 Alignment Evaluation Techniques

Many of the algorithms and articles in Ontology Alignment context uses *Precision* and *Recall* or their harmonic mean, referred to as *F-Measure*, to evaluate the performance of a method [4]. Also in some articles, they are used to evaluate alignment metrics[12]. In such methods after aggregation of results attained from different metrics, and extraction of mappings – based on one of the methods mentioned in [4] – the resulting mappings are compared with actual results.

In [8] a method for evaluation of Ontology Alignment methods – *Accuracy* – is proposed. This quality metric is based upon user effort needed to transform a match result obtained automatically into the intended result.

$$Accuracy = Recall \times (2 - \frac{1}{Precision}) \tag{1}$$

2.2 Calculation of Compound Similarity

The work closest to ours is probably that of Marc Ehrig et al. [3]. In *APFEL* weights for each feature is calculated using *Decision Trees*. The user only has to provide some ontologies with known correct alignments. The learned decision tree is used for aggregation and interpretation of the similarities.

3 Proposed Method

We first proposed a method to select appropriate metrics among existing set, and then introduce a method to combine them as a compound similarity. To use Precision, Recall, F-measure and Accuracy for metrics evaluation, it is needed to do mapping extraction. It depends on the definition of a *Threshold* value and the approach for extracting as well as on some defined constraints. Such dependencies results in in-appropriateness of current evaluation methods, although methods like what defines in [12] used to compare quality of metrics. We propose a new method for evaluation of metrics and creating a compound metric from some of them, featuring independent of mapping extraction phase, using learning.

Usually String and Linguistic based metrics are more influential than others and therefore if we want to select some metrics among existing metrics, most of the selected ones are linguistic which results in lower performance and flexibility of algorithm on different inputs. Therefor as a input for the training set, a number of metrics with their associated category is considered. Categories are for example *String Metric*, *Linguistic Metric*, *Structural Metric* and so on. Proposed algorithm selects one metric from each category. Furthermore, to enforce the algorithm to use a specific metric we can define a new category and introduce the metric as the only member of it. Like other learning based methods, it needs an initial training phase. In this phase a train set - an ontology pair with actual mappings in them - is given to the algorithm.

3.1 Learning Phase

In our algorithm, selection of appropriate metrics and aggregation of them are done based on *Data Mining* Techniques.

3.1.1 Reduction to a Data Mining Problem

For a pair of Ontologies a table is created with rows showing comparison of an entity from first ontology to an entity from the second one. For each metric under consideration a column is created in such a table with values showing normalized metric value for the pair of entities. An additional column with true or false values shows the existence of actual mapping between the two entities is also considered.

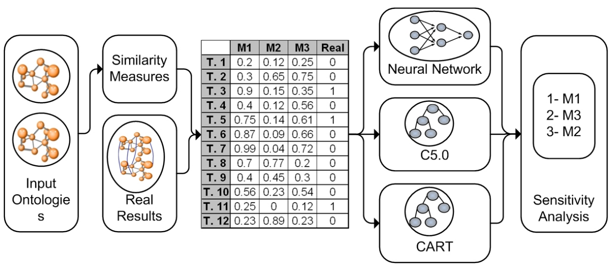

Figure 1. Proposed evaluation technique in detail

One table is created for each pair of Ontologies in the training set. Then all of such tables are aggregated in a single table. In this table, the column representing actual mapping value between a pair of entities is considered as target variable and the rest of columns are predictors. The problem now is a typical data mining problem and so we can apply classic data mining techniques to solve it. Fig. 1 shows the process. In this figure *Real Results* part shows the real mappings among entities of ontologies which are required during learning phase, and the *Sensitivity Analysis Rectangle* shows the results which are gain after sensitivity analysis, showing the appropriateness of metrics on the given train set.

3.1.2 Selection of Appropriate Metrics

In what following, we analysis the problem using Neural Networks as well as $CART^2$ and $C_{5.0}$ decision tress[6]. As mentioned before, columns of the table corresponding to values of metrics are considered as Predictors and the actual mapping value is the target variable. Fig. 1 shows the process. The aim is to find metrics having most influence in prediction of the target variable using Data Mining Models:

Neural Networks: *Sensitivity Analysis* for any problem is applied after a model has been constructed. With varying the values of input variables in the acceptable interval, the output variation is measured. With the interpretation of the output variation it is possible to recognize most influential input variable. After giving average value for each input variable to the model and measuring the output of the model, Sensitivity Analysis for each variable is done separately. To do this, the values of all variables except one in consideration are kept constant (their average value) and the model's response for minimum and maximum values of the variable in consideration are calculated. This process is repeated for all variables and then the variables with higher influence on variance of output are selected as most influential variables. For our problem it means that the metric having most variation on output during analysis is the most important metric.

Decision Trees: After creating the root node, in each iteration of the algorithm, a node is added to the decision tree. This process is repeated until the expansion of the tree is not possible anymore considering some predefined constraints. Selection of a variable as next node in the tree is done based on information theory concepts – in each repetition a variable with higher influence is selected among candidates. Therefore as a node is more near to the root, its corresponding variable has higher influence on the value of target variable. Hence from the constructed decision tree, it is possible to say that the metric in the root node has the highest influence.

3.1.3 Calculation of the Compound Metric

According to the results, and considering step 3-1, the problem is reduced to a Data Mining problem with the goal of finding an algorithm to compute target variable based on the predictor variables. In the Data Mining area several solutions have been proposed for these kind of problems. Among existing Data Mining solutions, we can refer to *CART* and $C_{5.0}$ [6] decision trees, A Priori for Association Rules generation [1] and Neural Networks [6]. Based on initial results among these methods, only *Neural Networks* has showed acceptable results. Neural Networks, have similar behavior with popular Alignment methods and they calculate Compound Similarity in the form of Weighted Sum with the weights is adjusted during learning.

Similar to the evaluation method, a table is constructed. As before, columns are the values selected metrics and an additional column records the target variable (0 or 1) showing the existence of a mapping between two entities. Now having such training samples a *Neural Network Model* is built. It is like a combined metric from the selected metrics which can be used as a new metric for the extraction phase.

[2] Classification And Regression Trees

4 Using the Proposed Method

To simplify the problem only String Based similarity metrics are considered. For the initial set of metrics we consider following metrics: the *Levenshtein* distance [7] which used the *Edit Distance* to match two strings, the *Needleman-Wunsch* distance[10], which assigns a different cost on the edit operations, the *Smith-Waterman* [11], which additionally uses an alphabet mapping to costs, the *Monge-Elkan* [9], which uses variable costs depending on the substring gaps between the words , the *Stoilos* similarity [12] which try to modify existing approaches for entities of an ontology, *Jaro-Winkler* similarity [5, 14], which counts the common characters between two strings even if they are misplaced by a "short" distance, and the *Sub-string* distance [4] which searches for the largest common substring. EON_{2004} [13] data set is used as the training set which is explained below: *Group* 1_{xx}*:* We only use test 103 from this group. Names of entities in this group is remaining without any changing and cause this group not to be a suitable data set for evaluation of string metrics. *Group* 2_{xx}*:* The reference ontology is compared with a modified one. Tests 204, 205, 221, 223 and 224 are used from this group. *Group* 3_{xx}*:* We use tests 302, 303 and 304 from this group. The reference ontology is compared with real-life ontologies. *All:* We merged all the data from described sets.

Each comparison of two strings is assigned a similarity degree. After collecting output for each metric, we evaluate them for each data set as it is described in Sect. 2. Fig 2 shows the results of applying *Sensitivity Analysis* on each test set after normalization. Levenshtein similarity is the most important one. Besides Sensitivity Analysis, Decision Tree models are

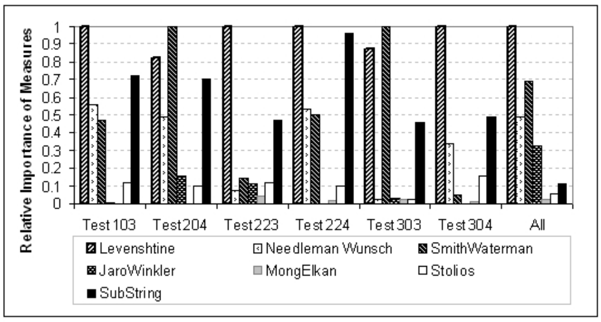

Figure 2. Evaluation of string metrics using Neural Networks

also used to confirm the results. In Table 1 we compare results of these techniques. All of three tests agree about importance of Levenshtein similarity on the test set. Neural Network chooses *Levenshtein* while $C_{5.0}$ and *CART* select it as second suitable metric. According to the presented algorithm and considering the fact that only one category is introduced as input, only Levenshtein is selected. In a more real situation the above steps are done for each category and one metric from each category is selected. *Levenshtein* and *Jaro-Winkler* are selected (from two imaginary categories). After creating a neural network with 4 layers and evaluation of the model on 3_{xx} test set, we got the convincing results.

Neural Network	CART	C5.0
Levenshtein	Jaro-Winkler	Needleman-Wunsch
SubString	Levenshtein	Levenshtein

Table 1. Most 2 important metrics

5 Conclusions

One advantage of the evaluation method is the uniform treatment of Similarity and Distance metrics so that we don't need to differentiate and process them separately. This is because in Data Mining evaluation, methods, there is no difference between a variable and a linear form of it. The alignment method can be improved when new metrics are introduced. In such cases it is only needed to add some new columns and do learning to adjust weights. Some of the researchers have emphasized on clustering and application of metrics for clusters as their future works. Another advantage of this method is that we can add cluster value as a new column to influence its importance for combination of metrics.

REFERENCES

[1] R. Agrawal, T. Imielinski, and et al, 'Mining association rules between sets of items in large databases', in *ACM SIGMOD Intl. Conf. Management of Data*, (May 1993).

[2] Paolo Bouquet, Marc Ehrig, and et al, 'Specification of a common framework for characterizing alignment', deliverable 2.2.1, Knowledge web NoE, (2004).

[3] Marc Ehrig, Staab Staab, and York Sure, 'Bootstrapping ontology alignment methods with APFEL', in *Proceedings of the 4th International Semantic Web Conference (ISWC-2005)*, eds., Yolanda Gil, Enrico Motta, and Richard Benjamins, Lecture Notes in Computer Science, pp. 186–200, (2005).

[4] Jérôme Euzenat, Thanh Le Bach, and et al, 'State of the art on ontology alignment', deliverable D2.2.3, Knowledge web NoE, (2004).

[5] M. Jaro, 'Probabilistic Linkage of Large Public Health Data Files', *Molecular Biology*, **14**, 491–498, (1995).

[6] Daniel T. Larose, *Discovering Knowledge In Data*, John Wiley and Sons, New Jersey, USA, 2005.

[7] Vladimir Levenshtein, 'Binary Codes Capable of Correcting Deletions, Insertions and Reversals', *Soviet Physics-Doklady*, **10**, 707–710, (August 1966).

[8] S. Melnik, H. Garcia-Molina, and et al, 'A versatile graph matching algorithm', in *Proceedings of ICDE*, (2002).

[9] Alvaro E. Monge and Charles P. Elkan, 'The Field-Matching Problem: Algorithm and Applications', in *Proceedings of the second international Conference on Knowledge Discovery and Data Mining*, (1996).

[10] S.B. Needleman and C.D. Wunsch, 'A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of two Proteins', *Molecular Biology*, **48**, (1970).

[11] T.F. Smith and M.S. Waterman, 'Identification of Common Molecular Subsequences', *Molecular Biology*, **147**, (1981).

[12] Giorgos Stoilos, Giorgos Stamou, and et al, 'A String Metric for Ontology Alignment', in *Proceedings of the ninth IEEE International Symposium on Wearable Computers*, pp. 624–237, (October 2005).

[13] Y. Sure, O. Corcho, and et al, eds. *Proceedings of the 3rd Evaluation of Ontology-based tools*, 2004.

[14] William E. Winkler, 'The State Record Linkage and Current Research Problems', Technical report, U. S. Bureau of the Census, Statistical Research Division, Room 3000-4, Bureau of the Census, Washington, DC, 20233-9100 USA, (1999).

Refining Ontologies via Pattern-based Clustering

Francesca A. Lisi and **Floriana Esposito** [1]

Abstract. In this paper we consider the problem of finding subconcepts of a known concept (reference concept) in a given ontology in the light of new knowledge coming from a data source. These subconcepts are discovered by looking for frequent association patterns between the reference concept and other concepts also occurring in the existing ontology. As an illustration, we report preliminary results obtained from the refinement of an $\mathcal{ALC}$ ontology with respect to DATALOG data extracted from the on-line CIA World Fact Book.

1 INTRODUCTION

Ontology Refinement is the adaptation of an existing ontology to a specific domain or the needs of a particular user [8]. In this paper we consider the problem of finding subconcepts of a known concept C_{ref}, called *reference concept*, in the existing ontology Σ in the light of new knowledge coming from a data source Π. We assume that a *concept* $\mathcal{C}$ consists of two parts: an *intension* $int(\mathcal{C})$ and an *extension* $ext(\mathcal{C})$. The former is an expression belonging to a logical language $\mathcal{L}$ whereas the latter is a set of objects that satisfy the former. More formally, given

- a reference concept $C_{ref} \in \Sigma$,
- a data set $\mathbf{r} = \Sigma \cup \Pi$,
- a language $\mathcal{L}$

our Ontology Refinement problem is to find a directed acyclic graph (DAG) $\mathcal{G}$ of concepts $\mathcal{C}_i$ such that (i) $int(\mathcal{C}_i) \in \mathcal{L}$ and (ii) $ext(\mathcal{C}_i) \subset ext(C_{ref})$. Note that C_{ref} is among both the concepts defined in Σ and the symbols of $\mathcal{L}$. Furthermore $ext(\mathcal{C}_i)$ relies on a notion of satisfiability of $int(\mathcal{C}_i)$ w.r.t. $\mathbf{r}$. Note that $\mathbf{r}$ includes Σ because in Ontology Refinement, as opposite to other forms of Ontology Learning such as Ontology Extraction (or Building), it is mandatory to consider the existing ontology and its existing connections.

A Knowledge Representation and Reasoning (KR&R) framework that turns out to be suitable for our problem is the one offered by the *hybrid system* $\mathcal{AL}$-log [2]. It allows for the specification of both relational and structural data: the former is based on DATALOG [1], the latter on $\mathcal{ALC}$ [11]. The integration of the two logical formalisms is provided by the so-called constrained DATALOG clause, i.e. a DATALOG clause with variables possibly constrained by concepts expressed in $\mathcal{ALC}$. Within this KR&R framework, the data set $\mathbf{r}$ is represented as a $\mathcal{AL}$-log knowledge base $\mathcal{B}$ and the language $\mathcal{L}$ contains expressions, called *$\mathcal{O}$-queries*, of the form

$$Q = q(X) \leftarrow \alpha_1, \ldots, \alpha_m \& X : C_{ref}, \gamma_1, \ldots, \gamma_n,$$

relating individuals of C_{ref} to individuals of other concepts (*task-relevant concepts*) also occurring in Σ. Thus, for a concept $\mathcal{C}$, $int(\mathcal{C})$ is an $\mathcal{O}$-query $Q \in \mathcal{L}$ and $ext(\mathcal{C})$ is the set $answerset(Q, \mathcal{B})$ of correct answers to Q w.r.t. $\mathcal{B}$. The DAG $\mathcal{G}$ is structured according to the *subset relation* between concept extensions.

The problem in hand can be considered as a case of tha form of unsupervised learning, known under the name of *Conceptual Clustering* [10], that aims at determining not only the clusters but also their descriptions expressed in some representation formalism. As a solution approach to the problem, we follow a recent trend in Cluster Analysis: using *frequent (association) patterns* as candidate clusters [13]. A frequent pattern is an intensional description, expressed in a language $\mathcal{L}$, of a subset of a given data set $\mathbf{r}$ whose cardinality exceeds a user-defined threshold (*minimum support*). Note that patterns can refer to multiple levels of description granularity (*multi-grained patterns*). In any case, a frequent pattern highlights a regularity in $\mathbf{r}$, therefore it can be considered as the clue of a data cluster. In the context of Ontology Refinement these clues are called *emerging concepts* because they are concepts whose only extension is determined. In [4] it has been proposed to extend [6] in order to provide a pattern-based approach to Conceptual Clustering.

The paper is organized as follows. Section 2 illustrates our approach to the problem. Section 3 reports a preliminary empirical evaluation of the approach. Section 4 concludes with final remarks and directions of future work.

2 PATTERN-BASED CLUSTERING

When faced with a pattern-based approach to Conceptual Clustering, the Ontology Refinement problem stated in Section 1 is decomposed in two subproblems:

I. discovery of frequent patterns in data
II. generation of clusters from frequent patterns

In particular, the subproblem I is actually a variant of frequent pattern discovery which aims at obtaining descriptions of the data set $\mathbf{r}$ at different levels of granularity [3]. Here $\mathbf{r}$ typically encompasses a taxonomy $\mathcal{T}$. More precisely, the problem of *frequent pattern discovery at l levels of description granularity*, $1 \leq l \leq maxG$, is to find the set $\mathcal{F}$ of all the frequent patterns expressible in a multi-grained language $\mathcal{L} = \{\mathcal{L}^l\}_{1 \leq l \leq maxG}$ and evaluated against $\mathbf{r}$ w.r.t. a set $\{minsup^l\}_{1 \leq l \leq maxG}$ of minimum support thresholds by means of the evaluation function $supp$. In this case, $P \in \mathcal{L}^l$ with support s is frequent in $\mathbf{r}$ if (i) $s \geq minsup^l$ and (ii) all ancestors of P w.r.t. $\mathcal{T}$ are frequent in $\mathbf{r}$.

The method proposed for solving one such decomposed problem extends the *levelwise search* method [9] for frequent pattern discovery with an additional post-processing step to solve the subproblem II. This method searches the space $(\mathcal{L}, \succeq)$ of patterns organized according to a generality order $\succeq$ in a breadth-first manner, starting

[1] Dipartimento di Informatica, Università degli Studi di Bari, Via Orabona 4, 70125 Bari, Italy, email: {lisi, esposito}@di.uniba.it

from the most general pattern in $\mathcal{L}$ and alternating candidate generation and candidate evaluation phases. The underlying assumption is that $\succeq$ is a quasi-order monotonic w.r.t. *supp*. For $\mathcal{L}$ being a multi-grained language of $\mathcal{O}$-queries, *supp* supplies the percentage of individuals of C_{ref} that satisfy an $\mathcal{O}$-query Q and $\succeq$ is based on the $\mathcal{B}$*-subsumption* relation [6]. It has been proved that $\succeq_{\mathcal{B}}$ is a quasi-order that fulfills the condition of monotonicity w.r.t. *supp* [6]. Also the search for patterns is depth-bounded (up to $maxD$).

The subproblem II concerns choosing a description for each cluster. In [5] it has been proposed a criterion obtained by combining two orthogonal biases: a language bias and a search bias. The language bias allows the user to define conditions on the form of $\mathcal{O}$-queries to be accepted as concept intensions. In particular, it is possible to state which is the minimum level of description granularity and whether (all) the variables must be ontologically constrained or not. The search bias allows the user to define a preference criterion based on $\mathcal{B}$-subsumption. In particular, it is possible to state whether the *most general description (m.g.d.)* or the *most specific description (m.s.d.)* w.r.t. $\succeq_{\mathcal{B}}$ has to be preferrred. Since $\succeq_{\mathcal{B}}$ is not a total order, it can happen that two patterns P and Q, belonging to the same language $\mathcal{L}$, can not be compared w.r.t. $\succeq_{\mathcal{B}}$. In this case, the m.g.d. (resp. m.s.d) of P and Q is the union (resp. conjunction) of P and Q.

Note that this method for Conceptual Clustering is *top-down* and *incremental* due to the features of the levelwise search. Also it is not hierarchical because it returns a DAG instead of a tree of concepts.

3 PRELIMINARY EXPERIMENTS

As an illustration, we report the results of four experiments conducted on the $\mathcal{AL}$-log knowledge base $\mathcal{B}_{\mathtt{CIA}}$ that has been obtained by adding DATALOG facts[2] extracted from the on-line 1996 CIA World Fact Book[3] to an $\mathcal{ALC}$ ontology $\Sigma_{\mathtt{CIA}}$ concerning the concepts `Country`, `EthnicGroup`, `Language`, and `Religion`. The parameter settings are: C_{ref} = `MiddleEastCountry`, $maxD = 5$, $maxG = 3$, $minsup^1 = 20\%$, $minsup^2 = 13\%$, and $minsup^3 = 10\%$. Thus each of them started from the same set $\mathcal{F}$ of 53 frequent patterns out of 99 candidate patterns.

Case for $l \geq 2$. The first two experiments both require the descriptions to have all the variables ontologically constrained by concepts from the second granularity level on. When the m.g.d. criterion is adopted, the procedure of graph building returns the following twelve concepts:

$\mathcal{C}'_0 \in \mathcal{F}^1_1$
`q(A)` ← `A:MiddleEastCountry`
{`ARM`, `BRN`, `IR`, `IRQ`, `IL`, `JOR`, `KWT`, `RL`, `OM`, `Q`, `SA`, `SYR`, `TR`, `UAE`, `YE`}

$\mathcal{C}'_1 \in \mathcal{F}^2_3$
`q(A)` ← `believes(A,B)` &
`A:MiddleEastCountry`, `B:MonotheisticReligion`
{`ARM`, `BRN`, `IR`, `IRQ`, `IL`, `JOR`, `KWT`, `RL`, `OM`, `Q`, `SA`, `SYR`, `TR`, `UAE`}

$\mathcal{C}'_2 \in \mathcal{F}^2_3$
`q(A)` ← `speaks(A,B)` &
`A:MiddleEastCountry`, `B:AfroAsiaticLanguage`
{`IR`, `SA`, `YE`}

$\mathcal{C}'_3 \in \mathcal{F}^2_3$
`q(A)` ← `speaks(A,B)` &
`A:MiddleEastCountry`, `B:IndoEuropeanLanguage`
{`ARM`, `IR`}

$\mathcal{C}'_4 \in \mathcal{F}^2_5$
`q(A)` ← `speaks(A,B)`, `believes(A,C)` &
`A:MiddleEastCountry`,
`B:AfroAsiaticLanguage`, `C:MonotheisticReligion`
{`IR`, `SA`}

$\mathcal{C}'_5 \in \mathcal{F}^2_5$
`q(A)` ← `believes(A,B)`, `believes(A,C)` &
`A:MiddleEastCountry`,
`B:MonotheisticReligion`, `C:MonotheisticReligion`
{`BRN`, `IR`, `IRQ`, `IL`, `JOR`, `RL`, `SYR`}

$\mathcal{C}'_6 \in \mathcal{F}^3_3$
`q(A)` ← `believes(A,'Druze')` & `A:MiddleEastCountry`
{`IL`, `SYR`}

$\mathcal{C}'_7 \in \mathcal{F}^3_3$
`q(A)` ← `believes(A,B)` &
`A:MiddleEastCountry`, `B:JewishReligion`
{`IR`, `IL`, `SYR`}

$\mathcal{C}'_8 \in \mathcal{F}^3_3$
`q(A)` ← `believes(A,B)` &
`A:MiddleEastCountry`, `B:ChristianReligion`
{`ARM`, `IR`, `IRQ`, `IL`, `JOR`, `RL`, `SYR`}

$\mathcal{C}'_9 \in \mathcal{F}^3_3$
`q(A)` ← `believes(A,B)` &
`A:MiddleEastCountry`, `B:MuslimReligion`
{`BRN`, `IR`, `IRQ`, `IL`, `JOR`, `KWT`, `RL`, `OM`, `Q`, `SA`, `SYR`, `TR`, `UAE`}

$\mathcal{C}'_{10} \in \mathcal{F}^3_5$
`q(A)` ← `believes(A,B)`, `believes(A,C)` &
`A:MiddleEastCountry`,
`B:ChristianReligion`, `C:MuslimReligion`
{`IR`, `IRQ`, `IL`, `JOR`, `RL`, `SYR`}

$\mathcal{C}'_{11} \in \mathcal{F}^3_5$
`q(A)` ← `believes(A,B)`, `believes(A,C)` &
`A:MiddleEastCountry`,
`B:MuslimReligion`, `C:MuslimReligion`
{`BRN`, `IR`, `SYR`}

organized in the DAG $\mathcal{G}'_{\mathtt{CIA}}$. They are numbered according to the chronological order of insertion in $\mathcal{G}'_{\mathtt{CIA}}$ and annotated with information of the generation step. From a qualitative point of view, concepts $\mathcal{C}'_2$[4] and $\mathcal{C}'_9$ well characterize Middle East countries. Armenia (`ARM`), as opposite to Iran (`IR`), does not fall in these concepts. It rather belongs to the weaker characterizations $\mathcal{C}'_3$ and $\mathcal{C}'_8$. This proves that our procedure performs a 'sensible' clustering. Indeed Armenia is a well-known borderline case for the geo-political concept of Middle East, though the Armenian is usually listed among Middle Eastern ethnic groups. Modern experts tend nowadays to consider it as part of Europe, therefore out of Middle East. But in 1996 the on-line CIA World Fact Book still considered Armenia as part of Asia.

When the m.s.d. criterion is adopted, the intensions for the concepts $\mathcal{C}'_4$, $\mathcal{C}'_6$ and $\mathcal{C}'_7$ change as follows:

$\mathcal{C}'_4 \in \mathcal{F}^2_5$
`q(A)` ← `speaks(A,B)`, `believes(A,C)` &
`A:MiddleEastCountry`,
`B:ArabicLanguage`, `C:MuslimReligion`
{`IR`, `SA`}

[2] `http://www.dbis.informatik.uni-goettingen.de/Mondial/mondial-rel-facts.flp`
[3] `http://www.odci.gov/cia/publications/factbook/`

[4] $\mathcal{C}'_2$ is less populated than expected because $\mathcal{B}_{\mathtt{CIA}}$ does not provide facts on the languages spoken for all countries.

$\mathcal{C}'_6 \in \mathcal{F}^3_3$
`q(A)` $\leftarrow$ `believes(A,'Druze')`, `believes(A,B)`,
`believes(A,C)`, `believes(A,D)` &
`A:MiddleEastCountry`, `B:JewishReligion`,
`C:ChristianReligion`, `D:MuslimReligion`
{`IL`, `SYR`}

$\mathcal{C}'_7 \in \mathcal{F}^3_3$
`q(A)` $\leftarrow$ `believes(A,B)`, `believes(A,C)`, `believes(A,D)` &
`A:MiddleEastCountry`, `B:JewishReligion`,
`C:ChristianReligion`, `D:MuslimReligion`
{`IR`, `IL`, `SYR`}

In particular $\mathcal{C}'_6$ and $\mathcal{C}'_7$ look quite overfitted to data. Yet overfitting allows us to realize that what distinguishes Israel (`IL`) and Syria (`SYR`) from Iran is just the presence of Druze people.

Case for $l \geq 3$. The other two experiments further restrict the conditions of the language bias specification. Here only descriptions with variables constrained by concepts of granularity from the third level on are considered. When the m.g.d. criterion is adopted, the procedure for graph building returns the following nine concepts:

$\mathcal{C}''_0 \in \mathcal{F}^1_1$
`q(A)` $\leftarrow$ `A:MiddleEastCountry`
{`ARM`, `BRN`, `IR`, `IRQ`, `IL`, `JOR`, `KWT`, `RL`, `OM`, `Q`, `SA`, `SYR`, `TR`, `UAE`, `YE`}

$\mathcal{C}''_1 \in \mathcal{F}^3_3$
`q(A)` $\leftarrow$ `speaks(A,B)` &
`A:MiddleEastCountry`, `B:ArabicLanguage`
{`IR`, `SA`, `YE`}

$\mathcal{C}''_2 \in \mathcal{F}^3_3$
`q(A)` $\leftarrow$ `believes(A,'Druze')` & `A:MiddleEastCountry`
{`IL`, `SYR`}

$\mathcal{C}''_3 \in \mathcal{F}^3_3$
`q(A)` $\leftarrow$ `believes(A,B)` &
`A:MiddleEastCountry`, `B:JewishReligion`
{`IR`, `IL`, `SYR`}

$\mathcal{C}''_4 \in \mathcal{F}^3_3$
`q(A)` $\leftarrow$ `believes(A,B)` &
`A:MiddleEastCountry`, `B:ChristianReligion`
{`ARM`, `IR`, `IRQ`, `IL`, `JOR`, `RL`, `SYR`}

$\mathcal{C}''_5 \in \mathcal{F}^3_3$
`q(A)` $\leftarrow$ `believes(A,B)` &
`A:MiddleEastCountry`, `B:MuslimReligion`
{`BRN`, `IR`, `IRQ`, `IL`, `JOR`, `KWT`, `RL`, `OM`, `Q`, `SA`, `SYR`, `TR`, `UAE`}

$\mathcal{C}''_6 \in \mathcal{F}^3_5$
`q(A)` $\leftarrow$ `speaks(A,B)`, `believes(A,C)` &
`A:MiddleEastCountry`,
`B:ArabicLanguage`, `C:MuslimReligion`
{`IR`, `SA`}

$\mathcal{C}''_7 \in \mathcal{F}^3_5$
`q(A)` $\leftarrow$ `believes(A,B)`, `believes(A,C)` &
`A:MiddleEastCountry`,
`B:ChristianReligion`, `C:MuslimReligion`
{`IR`, `IRQ`, `IL`, `JOR`, `RL`, `SYR`}

$\mathcal{C}''_8 \in \mathcal{F}^3_5$
`q(A)` $\leftarrow$ `believes(A,B)`, `believes(A,C)` &
`A:MiddleEastCountry`,
`B:MuslimReligion`, `C:MuslimReligion`
{`BRN`, `IR`, `SYR`}

organized in a DAG $\mathcal{G}''_{\mathtt{CIA}}$ which partially reproduces $\mathcal{G}'_{\mathtt{CIA}}$. Note that the stricter conditions set in the language bias cause two concepts occurring in $\mathcal{G}'_{\mathtt{CIA}}$ not to appear in $\mathcal{G}''_{\mathtt{CIA}}$: the scarsely significant $\mathcal{C}'_1$ and the quite interesting $\mathcal{C}'_3$.

When the m.s.d. condition is chosen, the intensions for the concepts $\mathcal{C}''_2$ and $\mathcal{C}''_3$ change analogously to $\mathcal{C}'_6$ and $\mathcal{C}'_7$.

4 CONCLUSIONS AND FUTURE WORK

Ontology Refinement can be considered as the problem of contextualizing an input ontology. In our case, context is conveyed by task-relevant concepts and is attached to the reference concept by discovering strong associations between the reference concepts and the task-relevant concepts w.r.t. the input ontology. The idea of applying association rules in Ontology Learning has been already investigated in [7]. Yet there are several differences: [7] is conceived for Ontology Extraction instead of Ontology Refinement, uses generalized association rules (bottom-up search) instead of multi-level association rules (top-down search), adopts propositional logic instead of First Order Logic. Also our work has contact points with Vrain's proposal [12] of a top-down incremental but distance-based method for Conceptual Clustering in a mixed object-logical representation.

For the future we plan to extensively evaluate our approach. Experiments will show, among the other things, how emerging concepts depend on the minimum support thresholds set for the stage of frequent pattern discovery.

References

[1] S. Ceri, G. Gottlob, and L. Tanca, *Logic Programming and Databases*, Springer, 1990.

[2] F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf, '$\mathcal{AL}$-log: Integrating Datalog and Description Logics', *Journal of Intelligent Information Systems*, **10**(3), 227–252, (1998).

[3] J. Han and Y. Fu, 'Mining multiple-level association rules in large databases', *IEEE Transactions on Knowledge and Data Engineering*, **11**(5), (1999).

[4] F.A. Lisi, 'A Pattern-Based Approach to Conceptual Clustering in FOL.', in *Conceptual Structures: Inspiration and Application*, eds., H. Schärfe, P. Hitzler, and P. Øhrstrøm, volume 4068 of *Lecture Notes in Artificial Intelligence*, 346–359, Springer, (2006).

[5] F.A. Lisi and F. Esposito, 'Two Orthogonal Biases for Choosing the Intensions of Emerging Concepts', in *ECAI 2006*. IOS Press, (2006).

[6] F.A. Lisi and D. Malerba, 'Inducing Multi-Level Association Rules from Multiple Relations', *Machine Learning*, **55**, 175–210, (2004).

[7] A. Maedche and S. Staab, 'Discovering Conceptual Relations from Text', in *Proceedings of the 14th European Conference on Artificial Intelligence*, ed., W. Horn, pp. 321–325. IOS Press, (2000).

[8] A. Maedche and S. Staab, 'Ontology Learning', in *Handbook on Ontologies*, eds., S. Staab and R. Studer, Springer, (2004).

[9] H. Mannila and H. Toivonen, 'Levelwise search and borders of theories in knowledge discovery', *Data Mining and Knowledge Discovery*, **1**(3), 241–258, (1997).

[10] R.S. Michalski and R.E. Stepp, 'Learning from observation: Conceptual clustering', in *Machine Learning: an artificial intelligence approach*, eds., R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, volume I, Morgan Kaufmann, San Mateo, CA, (1983).

[11] M. Schmidt-Schauss and G. Smolka, 'Attributive concept descriptions with complements', *Artificial Intelligence*, **48**(1), 1–26, (1991).

[12] C. Vrain, 'Hierarchical conceptual clustering in a first order representation.', in *Foundations of Intelligent Systems*, eds., Z.W. Ras and M. Michalewicz, volume 1079 of *Lecture Notes in Computer Science*, 643–652, Springer, (1996).

[13] H. Xiong, M. Steinbach, A. Ruslim, and V. Kumar, 'Characterizing pattern based clustering', Technical Report TR 05-015, Dept. of Computer Science and Engineering, University of Minnesota, Minneapolis, USA, (2005).

Reasoning with Fuzzy Ontologies

Yanhui Li[1] and **Baowen Xu**[1] and **Jianjiang Lu**[2] and **Dazhou Kang**[1]

Abstract. By the development of Semantic Web, increasing demands for vague information representation have triggered a mass of theoretical and applied researches of fuzzy ontologies, whose main logical infrastructures are fuzzy description logics. However, current tableau algorithms can not supply complete reasoning support within fuzzy ontology: reasoning with general TBox is still a difficult problem in fuzzy description logics. The main trouble is that fuzzy description logics adopt fuzzy models with continuous but not discrete membership degrees. In this paper, we propose a novel semantical discretization to discretize membership degrees in fuzzy description logic $\mathcal{FSHIN}$. Based on this discretization, we design discrete tableau algorithms to achieve reasoning with general TBox.

1 Introduction

The Semantic Web stands for the idea of a future Web, in which information is given well-defined meaning, better enabling intelligent Web information processing [1]. In the Semantic Web, ontology is a crucial knowledge representation model to express a shared understanding of information between users and machines. Along with the evolvement from current Web to the Semantic Web, the management of ill-structured, ill-defined or imprecise information plays a more and more important role in applications of the Semantic Web [13]. This trend calls for ontologies with capability to deal with uncertainty. However, classical DLs, as the logical foundation of ontologies, are two-value-based languages. The need for expressing uncertainty in the Semantic Web has triggered extending classical DLs with fuzzy capabilities, yielding Fuzzy DLs (FDLs for short). Straccia proposed a representative fuzzy extension $\mathcal{FALC}$ of DL $\mathcal{ALC}$, in which fuzzy semantics is introduced to interpret concepts and roles as fuzzy sets [11]. Following researchers extended $\mathcal{FALC}$ with more complex constructions: $\mathcal{FALCQ}$ [6] with qualified number restriction , $\mathcal{FSI}$ [7] with transitive and inverse role, and $\mathcal{FSHIN}$ [8], a extension of $\mathcal{FSI}$ with role hierarchy and unqualified number restriction. Stoilos et al introduced Straccia's fuzzy framework into OWL, hence getting a fuzzy ontology language $\mathcal{FSHOIN}$, by which fuzzy ontologies are coded as FDL knowledge bases [9].

Though the fuzzy DLs have done a lot, to our best knowledge, reasoning with general TBox in FDLs is still a difficult problem [8]. Current tableau algorithms in FDLs are applied to achieve reasoning without TBox or with acyclic TBox [7, 8, 11], that limits reasoning support within fuzzy ontologies. The main trouble in reasoning with general TBox is that fuzzy interpretations $\mathcal{I}$ map concepts C into membership degree functions $C^{\mathcal{I}}()$ w.r.t domain $\Delta^{\mathcal{I}}$: $\Delta^{\mathcal{I}} \to [0,1]$, where the value domain [0,1] is continuous. In [4], we represented a novel semantical discretization technique to enable translation of membership degree values from $continuous$ ones into $discrete$ ones. In this paper, we will extend this discretization technique into $\mathcal{FSHIN}$; and based on it, we will design a discrete tableau algorithm for reasoning with general TBox in $\mathcal{FSHIN}$. Since nominals should not be fuzzyfied, our discrete tableau algorithms for $\mathcal{SHIN}$, together with reasoning technique to deal with nominals in crisp DLs [3], can be extended to provide a tableau algorithm for general TBox in $\mathcal{FSHOIN}$, that will achieve complete reasoning within fuzzy ontologies.

[1] Department of Computer Science and Engineering, Southeast University, Nanjing 210096, P. R. China. email: yanhuili,bwxu,swws@seu.edu.cn
[2] Institute of Command Automation, PLA University of Science and Technology, Nanjing 210007, China. email:jjlu@seu.edu.cn

2 Logical Infrastructure of Fuzzy Ontologies

Let $\mathrm{N_C}$ be a set of concept names (A), $\mathrm{N_R}$ a set of role names (R) with a subset $\mathrm{N_R}^+$ of transitive role names and $\mathrm{N_I}$ a set of individual names (a). $\mathcal{FSHIN}$ roles are either role names $R \in \mathrm{N_R}$ or their inverse roles R^-. To avoid R^{--}, we use $\mathrm{Inv}(R)$ to denote the inverse role of R. $\mathcal{FSHIN}$ concepts C, D are inductively defined with the application of $\mathcal{FSHIN}$ concept constructors in the following syntax rules:

$$C, D :: \top|\bot|A|\neg C|C \sqcap D|C \sqcup D|\exists R.C|\forall R.C| \geq pR| \leq pR$$

Since concepts and roles in $\mathcal{FSHIN}$ are considered as fuzzy sets, the semantics of concepts and roles are defined in terms of fuzzy interpretations $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a nonempty domain, and $\cdot^{\mathcal{I}}$ is an interpretation function mapping individuals a into $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$; concept (role) names A (R) into membership functions $A^{\mathcal{I}}(R^{\mathcal{I}}) : \Delta^{\mathcal{I}}$ $(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \to [0,1]$. And for any transitive role name $R \in \mathrm{N_R}^+$, $\mathcal{I}$ satisfies $\forall d, d' \in \Delta^{\mathcal{I}}$, $R^{\mathcal{I}}(d,d') \geq \sup_{x \in \Delta^{\mathcal{I}}}\{\min(R^{\mathcal{I}}(d,x), R^{\mathcal{I}}(x,d'))\}$. Furthermore, $\cdot^{\mathcal{I}}$ satisfies the following conditions for complex concepts and roles built by concept and role constructors: for any $d, d' \in \Delta^{\mathcal{I}}$

$$\begin{aligned}
\top^{\mathcal{I}}(d) &= 1 \\
\bot^{\mathcal{I}}(d) &= 0 \\
(\neg C)^{\mathcal{I}}(d) &= 1 - C^{\mathcal{I}}(d) \\
(C \sqcap D)^{\mathcal{I}}(d) &= \min\{C^{\mathcal{I}}(d), D^{\mathcal{I}}(d)\} \\
(C \sqcup D)^{\mathcal{I}}(d) &= \max\{C^{\mathcal{I}}(d), D^{\mathcal{I}}(d)\} \\
(\exists R.C)^{\mathcal{I}}(d) &= \sup{}_{d' \in \Delta^{\mathcal{I}}}\{\min(R^{\mathcal{I}}(d,d'), C^{\mathcal{I}}(d'))\} \\
(\forall R.C)^{\mathcal{I}}(d) &= \inf{}_{d' \in \Delta^{\mathcal{I}}}\{\max(1 - R^{\mathcal{I}}(d,d'), C^{\mathcal{I}}(d'))\} \\
(\geq pR)^{\mathcal{I}}(d) &= \sup{}_{d_1, d_2, \ldots, d_p \in \Delta^{\mathcal{I}}}\{\min_1^p(R^{\mathcal{I}}(d,d_i))\} \\
(\leq pR)^{\mathcal{I}}(d) &= \inf{}_{d_1, d_2, \ldots, d_{p+1} \in \Delta^{\mathcal{I}}}\{\max_1^{p+1}(1 - R^{\mathcal{I}}(d,d_i))\} \\
(R^-)^{\mathcal{I}}(d,d') &= R^{\mathcal{I}}(d',d)
\end{aligned}$$

A $\mathcal{FSHIN}$ knowledge base (KB) $\mathcal{K}$ is a triple $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, where $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$ are $\mathcal{FSHIN}$ TBox, RBox and ABox. The syntax and semantics of axioms in them are given in table 1. An interpretation $\mathcal{I}$ satisfies an axiom if it satisfies corresponding semantics restriction given in table 1. $\mathcal{I}$ satisfies (is a fuzzy model of) a KB $\mathcal{K}$, iff $\mathcal{I}$ satisfies any axiom in $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$. $\mathcal{K}$ is satisfiable iff it has a fuzzy model. In this paper, we will propose a discrete tableau algorithm to decide satisfiability of $\mathcal{FSHIN}$ KBs, which is based on the "semantical discretization" discussed in the following section.

Table 1. Syntax and semantics of $\mathcal{FSHIN}$ axioms

	Syntax	Semantics
TBox $\mathcal{T}$	$C \sqsubseteq D$	$\forall d \in \Delta^{\mathcal{I}}, C^{\mathcal{I}}(d) \leq D^{\mathcal{I}}(d)$
RBox $\mathcal{R}$	$R \sqsubseteq P$	$\forall d, d' \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(d,d') \leq P^{\mathcal{I}}(d,d')$
ABox $\mathcal{A}$	$a : C \bowtie n$	$C^{\mathcal{I}}(a^{\mathcal{I}}) \bowtie n$
	$\langle a,b \rangle : R \bowtie$	$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \bowtie n$
	$a \neq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

C and D (R and P) are concepts (roles); $a, b \in \mathrm{N_I}$; $\bowtie \in \{\geq, >, \leq, <\}$; $n \in$ [0,1].

3 Semantical Discretization in $\mathcal{FSHIN}$

For any fuzzy model of $\mathcal{FSHIN}$ KBs, we discretize it into a special model, in which any value of membership degree functions belongs to a given discrete degree set S. And we call it a discrete model within S. Let us now proceed formally in the creation of S. Let N_d be the set of degrees appearing in ABox $N_d = \{n|\alpha \bowtie n \in \mathcal{A}\}$. From N_d, we define the degree closure $N_d^* = \{0, 0.5, 1\} \cup N_d \cup \{n|1-n \in N_d\}$ and order degrees in ascending order: $N_d^* = \{n_0, n_1, \ldots, n_s\}$, where for any $0 \leq i \leq s$, $n_i < n_{i+1}$. For any two back-to-back elements n_i, $n_{i+1} \in N_d^*$, we insert their median $m_{i+1} = (n_i + n_{i+1})/2$ to get $S = \{n_0, m_1, n_1, \ldots, n_{s-1}, m_s, n_s\}$. We call S a discrete degree set w.r.t $\mathcal{K}$. Obviously for any $1 \leq i \leq s$, $m_i + m_{s+1-i} = 1$ and $n_{i-1} < m_i < n_i$.

Theorem 1 *For any $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ and any discrete degree set S w.r.t $\mathcal{K}$, iff $\mathcal{K}$ has a fuzzy model, it has a discrete model within S.*

Proof. Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ be a fuzzy model of $\mathcal{K}$ and the degree set $S = \{n_0, m_1, n_1, \ldots, n_{s-1}, m_s, n_s\}$. Consider a translation function $\varphi() : [0,1] \to S$:

$$\varphi(x) = \begin{cases} n_i & \text{if } x = n_i \\ m_i & \text{if } n_{i-1} < x < n_i \end{cases}$$

Based on $\varphi()$, we will construct a discrete model $\mathcal{I}_c = \langle \Delta^{\mathcal{I}_c}, \cdot^{\mathcal{I}_c} \rangle$ within S from $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$:

- The interpretation domain $\Delta^{\mathcal{I}_c}$ is defined as: $\Delta^{\mathcal{I}_c} = \Delta^{\mathcal{I}}$;
- The interpretation function $\cdot^{\mathcal{I}_c}$ is defined as: for any individual name a, $a^{\mathcal{I}_c} = a^{\mathcal{I}}$; for any concept name A and any role name R: $A^{\mathcal{I}_c}() = \varphi(A^{\mathcal{I}}())$ and $R^{\mathcal{I}_c}() = \varphi(R^{\mathcal{I}}())$.

1. For any concept C and role R and any $d, d' \in \Delta^{\mathcal{I}_c}$, we show, on induction on the structure of C and R, that $C^{\mathcal{I}_c}(d) = \varphi(C^{\mathcal{I}}(d))$ and $R^{\mathcal{I}_c}(d,d') = \varphi(R^{\mathcal{I}}(d,d'))$:
   - $\geq pR$: $(\geq pR)^{\mathcal{I}}(d) = \sup_{d_1,d_2,\ldots,d_p \in \Delta^{\mathcal{I}}} \{\min_1^p(R^{\mathcal{I}}(d,d_i))\}$. Let $f(d') = R^{\mathcal{I}}(d,d')$, and $f^*(d') = \varphi(f(d))$. Assume there are p elements $d_1^*, d_2^*, \ldots, d_p^*$ with the maximum value of $f()$: for any other d' in $\Delta^{\mathcal{I}}$, $f(d_i^*) \geq f(d')$. Obviously from the property of $\varphi()$, for any other d' in $\Delta^{\mathcal{I}_c}$, $f^*(d_i^*) = \varphi(f(d_i^*)) \geq \varphi(f(d)) = f^*(d')$. Then we get
   $$\begin{aligned}(\geq pR)^{\mathcal{I}_c}(d) &= \sup_{d_1,d_2,\ldots,d_p \in \Delta^{\mathcal{I}_c}} \{\min_1^p(f^*(d_i))\} \\ &= \min_1^p(f^*(d_i^*)) = \varphi(\min_1^p(f(d_i^*))) \\ &= \varphi(\sup_{d_1,d_2,\ldots,d_p \in \Delta^{\mathcal{I}}} \{\min_1^p(R^{\mathcal{I}}(d,d_i))\}) \\ &= \varphi((\geq pR)^{\mathcal{I}}(d))\end{aligned}$$
2. We show $\mathcal{I}_c$ is a fuzzy model of $\mathcal{K}$.
   - $C \sqsubseteq D \in \mathcal{T}$: Obviously, $\forall d \in \Delta^{\mathcal{I}} = \Delta^{\mathcal{I}_c}, C^{\mathcal{I}}(d) \leq D^{\mathcal{I}}(d)$. And from 1, for any concept C, $C^{\mathcal{I}_c}(d) = \varphi(C^{\mathcal{I}}(d))$. Therefore, $C^{\mathcal{I}_c}(d) = \varphi(C^{\mathcal{I}}(d)) \leq \varphi(D^{\mathcal{I}}(d)) = D^{\mathcal{I}_c}(d)$;

4 Discrete Tableau Algorithms for $\mathcal{FSHIN}$

For a KB $\mathcal{K}$, let $\mathrm{R}_\mathcal{K}$ and $\mathrm{O}_\mathcal{K}$ be the sets of roles and individuals appearing in $\mathcal{K}$, and $\mathrm{sub}(\mathcal{K})$ the set of sub-concepts of all concepts in $\mathcal{K}$. We also introduce $\mathrm{Trans}(R)$ as a boolean value to tell whether R is transitive, $\rhd$ and $\lhd$ as two placeholders for the inequalities $\geq$, $>$ and $\leq$, $<$, and the symbols $\bowtie^-$, $\rhd^-$ and $\lhd^-$ to denote their reflections. A discrete tableau T for $\mathcal{K}$ within a degree set S is a quadruple: $\langle \mathcal{O}, \mathcal{L}, \mathcal{E}, \mathcal{V} \rangle$, where

- $\mathcal{O}$: a nonempty set of nodes;
- $\mathcal{L}$:$\mathcal{O} \to 2^M$, $M = \mathrm{sub}(\mathcal{K}) \times \{\geq, >, \leq, <\} \times S$;
- $\mathcal{E}$: $\mathrm{R}_\mathcal{K} \to 2^Q$, $Q = \{\mathcal{O} \times \mathcal{O}\} \times \{\geq, >, \leq, <\} \times S$;
- $\mathcal{V}$:$\mathrm{O}_\mathcal{K} \to \mathcal{O}$, maps any individual into a corresponding node in $\mathcal{O}$.

From the definition of T, each node d is labelled with a set $\mathcal{L}(d)$ of degree triples: $\langle C, \bowtie, n \rangle$, which denotes the membership degree of d being an instance of $C \bowtie n$. In a discrete tableau T, for any $d, d' \in \mathcal{O}$, $a, b \in \mathrm{O}_\mathcal{K}$, $C, D \in \mathrm{sub}(\mathcal{K})$ and $R \in \mathrm{R}_\mathcal{K}$, the following conditions, a extension of tableau conditions in dealing without TBox [8] by adding KB conditions and NNF conditions, must hold:

KB condition: If $C \sqsubseteq D \in \mathcal{T}$, then there must be some $n \in S$ with $\langle C, \leq, n \rangle$ and $\langle D, \geq, n \rangle$ in $\mathcal{L}(d)$.
NNF condition: If $\langle C, \bowtie, n \rangle \in \mathcal{L}(d)$, then $\langle \mathrm{nnf}(\neg C), \bowtie^-, 1 - n \rangle \in \mathcal{L}(d)$. Here we use $\mathrm{nnf}(\neg C)$ to denote the equivalent form of $\neg C$ in Negation Normal Form (NNF).

Theorem 2 *For any $\mathcal{K} =< \mathcal{T}, \mathcal{R}, \mathcal{A} >$ and any discrete degree set S w.r.t $\mathcal{K}$, $\mathcal{K}$ has a discrete model within S iff it has a discrete tableau T within S.*

From theorem 1 and 2, an algorithm that constructs a discrete tableau of $\mathcal{K}$ within S can be considered as a decision procedure for the satisfiability of $\mathcal{K}$. The discrete tableau algorithm works on a completion forest $\mathrm{F}_\mathcal{K}$ with a set $S^{\neq}$ to denote "$\neq$" relation between nodes. The algorithm expands the forest $\mathrm{F}_\mathcal{K}$ either by extending $\mathcal{L}(x)$ for the current node x or by adding new leaf node y with expansion rules in table 2. A node y is called an R-successor of another node x and x is called a R-predecessor of y, if $\langle R, \bowtie, n \rangle \in \mathcal{L}(\langle x, y \rangle)$. Ancestor is the transitive closure of predecessor. And for any two connected nodes x and y, we define $D_R(x,y)$=$\{\langle \bowtie, n \rangle | P \sqsubseteq^* R, \langle P, \bowtie, n \rangle \in \mathcal{L}(\langle x, y \rangle)$ or $\langle \mathrm{Inv}(P), \bowtie, n \rangle \in \mathcal{L}(\langle y, x \rangle)\}$. If $D_R(x,y) \neq \emptyset$, y is called a R-neighbor of x.

The tableau algorithm initializes $\mathrm{F}_\mathcal{K}$ to contain a root node x_a for each individual $a \in \mathrm{O}_\mathcal{K}$ and labels x_a with $\mathcal{L}(x_a)$= $\{\langle C, \bowtie, n \rangle | a : C \bowtie n \in \mathcal{A}\}$; for any pair $\langle x_a, x_b \rangle$, $\mathcal{L}\langle x_a, x_b \rangle$=$\{\langle R, \bowtie, n \rangle | \langle a, b \rangle : R \bowtie n \in \mathcal{A}\}$; and for any $a \neq b \in \mathcal{A}$, $\langle x_a, x_b \rangle \in S^{\neq}$. As inverse role and number restriction are allowed in $\mathcal{SHIN}$, we make use of pairwise blocking technique [2] to ensure the termination and correctness of our tableau algorithm: a node x is directly blocked by its ancestor y iff (1) x is not a root node; (2) x and y have predecessors x' and y', such that $\mathcal{L}(x) = \mathcal{L}(y)$ and $\mathcal{L}(x') = \mathcal{L}(y')$ and $\mathcal{L}(\langle y', y \rangle) = \mathcal{L}(\langle x', x \rangle)$. A node x is indirectly blocked if its predecessor is blocked. A node x is blocked iff it is either directly or indirectly blocked. A completion forest $\mathrm{F}_\mathcal{K}$ is said to contain a clash, if for a node x in $\mathrm{F}_\mathcal{K}$, (1) $\mathcal{L}(x)$ contains two conjugated triples, or a mistake triple [4]; or (2) $\langle \geq pR, \lhd, n \rangle$ or $\langle \leq (p-1)R, \lhd^-, 1-n \rangle \in \mathcal{L}(x)$, and there are p nodes $y_1, y_2, \ldots y_p$ in $\mathrm{F}_\mathcal{K}$ with $\langle R, \rhd_i, m_i \rangle$, $\langle \rhd_i, m_i \rangle$ is conjugated with $\langle \lhd, n \rangle$ and for any two nodes y_i and y_j, $\langle y_i, y_j \rangle \in S^{\neq}$. A completion forest $\mathrm{F}_\mathcal{K}$ is clash-free if it does not contain a clash, and it is complete if none of the expansion rules are applicable.

Table 2. Expansion rules of discrete Tableau

Rule name	Description
KB rule:	if $C \sqsubseteq D \in \mathcal{T}$ and there is no n with $\langle C, \leq, n\rangle$ and $\langle D, \geq, n\rangle$ in $\mathcal{L}(x)$; then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\langle C, \leq, n\rangle\ \langle D, \geq, n\rangle\}$ for some $n \in S$.
The following rules are applied to nodes x which is not indirectly blocked.	
$\neg^{\bowtie}$ rule:	if $\langle C, \bowtie, n\rangle \in \mathcal{L}(x)$ and $\langle \text{nnf}(\neg C), \bowtie^{-}, n\rangle \notin \mathcal{L}(x)$; then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\langle \text{nnf}(\neg C), \bowtie^{-}, n\rangle\}$.
$\sqcap^{\rhd}$ rule:	if $\langle C \sqcap D, \rhd, n\rangle \in \mathcal{L}(x)$, and $\langle C, \rhd, n\rangle$ or $\langle D, \rhd, n\rangle \notin \mathcal{L}(x)$; then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\langle C, \rhd, n\rangle, \langle D, \rhd, n\rangle\}$.
$\sqcup^{\rhd}$ rule:	if $\langle C \sqcup D, \rhd, n\rangle \in \mathcal{L}(x)$, and $\langle C, \rhd, n\rangle, \langle D, \rhd, n\rangle \notin \mathcal{L}(x)$ then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{T\}$, for some $T \in \{\langle C, \rhd, n\rangle, \langle D, \rhd, n\rangle\}$
$\forall^{\rhd}$ rule:	if $\langle \forall R.C, \rhd, n\rangle \in \mathcal{L}(x)$, there is a R-neighbor y of x with $\langle \rhd', m\rangle \in D_R(x, y)$, which is conjugated with $\langle \rhd^{-}, 1-n\rangle$ and $\langle C, \rhd, n\rangle \notin \mathcal{L}(y)$; then $\mathcal{L}(y) \rightarrow \mathcal{L}(y) \cup \{\langle C, \rhd, n\rangle\}$.
$\forall^{+\rhd}$ rule:	if $\langle \forall P.C, \rhd, n\rangle \in \mathcal{L}(x)$, there is a R-neighbor y of x with $R \sqsubseteq^{*} P$, $\text{Trans}(R)$=True and $\langle \rhd', m\rangle \in D_R(x, y)$, $\langle \rhd', m\rangle$ is conjugated with $\langle \rhd^{-}, 1-n\rangle$ and $\langle \forall R.C, \rhd, n\rangle \notin \mathcal{L}(y)$; $\mathcal{L}(y) \rightarrow \mathcal{L}(y) \cup \{\langle \forall R.C, \rhd, n\rangle\}$.
$\leq p^{\rhd}$ rule:	if $\langle \leq pR, \rhd, n \in \mathcal{L}(x)$; there is $p+1$ R-successors $y_1, y_2, \ldots, y_{p+1}$ of x with $\langle R, \rhd_i, m_i\rangle \in \mathcal{L}(\langle x, y_i\rangle)$ and $\langle \rhd_i, m_i\rangle$ is conjugated with $\langle \lhd^{-}, 1-n\rangle$ for any $1 \leq i \leq p+1$; and $\langle y_i, y_j\rangle \notin S^{\neq}$ for some $1 \leq i < j \leq p+1$ then merge two nodes y_i and y_j into one : $\mathcal{L}(y_i) \rightarrow \mathcal{L}(y_i) \cup \mathcal{L}(y_j)$; $\forall x, \mathcal{L}(y_i, x) \rightarrow \mathcal{L}(y_i, x) \cup \mathcal{L}(y_j, x)$, $\langle y_j, x\rangle \in S^{\neq}$, add $\langle y_i, x\rangle$ in $S^{\neq}$
The following rules are applied to nodes x which is not blocked.	
$\exists^{\rhd}$ rule:	if $\langle \exists R.C, \rhd, n\rangle \in \mathcal{L}(x)$; there is not a R-neighbor y of x with $\langle \rhd, n\rangle \in D_R(x, y)$ and $\langle C, \rhd, n\rangle \in \mathcal{L}(y)$. then add a new node z with $\langle R, \rhd, n\rangle \in \mathcal{L}(\langle x, z\rangle)$ and $\langle C, \rhd, n\rangle \in \mathcal{L}(z)$.
$\geq pR^{\rhd}$ rule:	if $\langle \geq pR, \rhd, n\rangle \in \mathcal{L}(x)$, there are not p R-neighbors $y_1, y_2, \ldots, y_p$ of x with $\langle R, \rhd, n\rangle \in \mathcal{L}(\langle x, y_i\rangle)$ and for any $i \neq j$, $\langle y_i, y_j\rangle \in S^{\neq}$. then add p new nodes $z_1, z_2, \ldots, z_p$ with $\langle R, \rhd, n\rangle \in \mathcal{L}(\langle x, z_i\rangle)$ and for any two node z_i and z_j, add $\langle z_i, z_j\rangle$ in $S^{\neq}$.

Theorem 3 *For any $\mathcal{K} =< \mathcal{T}, \mathcal{R}, \mathcal{A} >$ and any discrete degree set S w.r.t $\mathcal{K}$, $\mathcal{K}$ has a discrete tableau within S iff the tableau algorithm can construct a complete and clash-free completion forest.*

5 Related Work

In FDLs area, we have introduced a lot of work in introduction, all that work are based on Straccia' fuzzification framework. Here we get into reasoning issue for fuzzy DLs. The first reasoning algorithm was represented in [10], and the soundness and completeness of it were proved in [11]. This algorithm is designed to reasoning with $\mathcal{FALC}$ acyclic TBox form. More in detail, it first adopted KB expansion [5] to eliminate acyclic TBox, then achieved reasoning without TBox. However, such expansion technique is not available for general TBox in FDLs. The following extension of $\mathcal{FALC}$ inherited this idea to design reasoning algorithm, so most of these extension are limited to dealing with empty or acyclic TBox. In general TBox cases, a noteworthy reasoning method is PTIME bounded translations from $\mathcal{FALCH}$ KBs into $\mathcal{ALCH}$ ones and reusing existing classical algorithm to achieve reasoning in fuzzy DLs [12]. This PTIME bounded translation can be considered as a result of researches on relationship between DLs and fuzzy DLs. It can not deal with $\langle a, b\rangle : R \lhd n$ in $\mathcal{A}$, as this assertion will be translated into role negation (that is not allowed in $\mathcal{ALC}$).

6 Conclusion

In this paper, we point out a novel semantical discretization to discretize membership degree values in fuzzy models of $\mathcal{FSHIN}$ KBs, hence yielding "discrete models". Based on this discretization technique, we design a discrete tableau algorithm to construct discrete tableaus, which are abstraction of discrete models. From the equivalence of existence between fuzzy models and discrete models, our algorithm is a decision procedure to achieve reasoning with general TBox in $\mathcal{FSHIN}$ KBs. Our work can be considered as a logical foundation to support reasoning with fuzzy ontologies.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila, 'The semantic web', *Scientific American*, **284**(5), 34–43, (2001).

[2] I. Horrocks and U. Sattler, 'A description logic with transitive and inverse roles and role hierarchies', *Journal of Logic and Computation*, **9**, 385–410, (1999).

[3] I. Horrocks and U. Sattler, 'A tableaux decision procedure for shoiq', in *Proceedings of 19th Int. Joint Conf. on Artificial Intelligence*, (2005).

[4] Y.H. Li, B.W. Xu, J.J. Lu, and D.Z. Kang, 'Discrete tableau Algorithms for fshi', in *Proceedings of 2006 International Workshop on Description Logics - DL2006*, The Lake District of the UK, (2006).

[5] B. Nebel, 'Computational complexity of terminological reasoning in back', *Artificial Intelligence*, **34**, 371–383, (1988).

[6] D. Sanchez and G. Tettamanzi, 'Generalizing quantification in fuzzy description logic', in *Proceedings of the 8th Fuzzy Days*, (2004).

[7] G. Stoilos, G. Stamou, V. Tzouvaras, J.Z. Pan, and I. Horrock, 'A Fuzzy Description Logic for Multimedia Knowledge Representation', in *Proc. of the International Workshop on Multimedia and the Semantic Web*, (2005).

[8] G. Stoilos, G. Stamou, V. Tzouvaras, J.Z. Pan, and I. Horrocks, 'The fuzzy description logic shin', in *Proceedings of International Workshop of OWL: Experiences and Directions*, Galway, (2005).

[9] G. Stoilos, G. Stamou, V. Tzouvaras, J.Z. Pan, and I. Horrocks, 'Fuzzy owl: Uncertainty and the semantic web', in *Proceedings of International Workshop of OWL: Experiences and Directions*, Galway, (2005).

[10] U. Straccia, 'A fuzzy description logic', in *Proceedings of AAAI-98, 15th National Conference on Artificial Intelligence*, pp. 594–599, Madison, Wisconsin, (1998).

[11] U. Straccia, 'Reasoning within fuzzy description logics', *Journal of Artificial Intelligence Researc*, **14**, 137–166, (2001).

[12] U. Straccia, 'Transforming fuzzy description logics into classical description logics', in *Proceedings of the 9th European Conference on Logics in Artificial Intelligence*, pp. 385–399, Lisbon, (2004).

[13] D.H. Widyantoro and J. Yen, 'Using fuzzy ontology for query refinement in a personalized abstract search engine', in *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, (2001).

Background knowledge for ontology construction

Blaz Fortuna and **Marko Grobelnik** and **Dunja Mladenic**[1]

Abstract. In this paper we describe a solution for incorporating background knowledge into the OntoGen system for semi-automatic ontology construction. This makes it easier for different users to construct different and more personalized ontologies for the same domain. To achieve this we introduce a word weighting schema to be used in the document representation. The weighting schema is learned based on the background knowledge provided by user. It is than used by OntoGen's machine learning and text mining algorithms.

1 INTRODUCTION

When using ontology-based techniques for knowledge management it is important for the ontology to capture the domain knowledge in a proper way. Very often different tasks and users require the knowledge to be encoded into ontology in different ways, depending on the task. For instance, the same document-database in a company may be viewed differently by marketing, management, and technical staff. Therefore it is crucial to develop techniques for incorporating user's background knowledge into ontologies.

In [4] we introduced a system called OntoGen for semi-automatic construction of topic ontologies. Topic ontology consists of a set of topics (or concepts) and a set of relations between the topics which best describe the data. The OntoGen system helps the user by discovering possible concepts and relations between them within the data.

In this paper we propose a method which extends OntoGen system so that the user can supervise the methods for concept discovery by providing background knowledge - his specific view on the data used by the text mining algorithms in the system.

To encode the background knowledge we require from the user to group documents into categories. These categories do not need to describe the data in details, the important thing is that they show to the system the user's view of the data - which documents are similar and which are different from the user's perspective. The process of manually marking the documents with categories is time consuming but can be significantly speeded up by the use of active learning [5], [8]. Another source of such labeled data could be popular online tagging services (e.g Del.icio.us) which allow the user to label the websites of his interests with labels he chose.

This paper is organized as follows. In Section 2 we introduce OntoGen system and in Section 3 we derive the algorithm for calculating word weights. We conclude the paper with some preliminary results in Section 4.

[1] Institute Jozef Stefan, Slovenia, email: {blaz.fortuna, marko.grobelnik, dunja.mladenic}@ijs.si

2 ONTOGEN

OntoGen [4] is a system for semi-automatic ontology construction, screenshot of the tool is presented in the Figure 1. Important part of OntoGen are methods for discovering concepts from a collection of documents. For the representation of the documents we use the well established bag-of-words representation which heavily relies on the weights associated with the words. The weights of the words are commonly calculated by so called TFIDF weighting. We argue that this provides just one of the possible views on the data and propose an alternative word weighting that takes into account the background knowledge which provides the user's view on the documents.

OntoGen discovers concepts using Latent Semantic Indexing (LSI) [3] and k-means clustering [6]. The LSI is a method for linear dimensionality reduction by learning an optimal sub-basis which approximates documents' bag-of-words vectors. The sub-basis vectors are treated as concepts. The k-means method discovers concepts by clustering the documents' bag-of-words vectors into k clusters where each cluster is treated as a concept.

Both methods heavily rely on the representation of the documents. Namely, the document representation provides the vectors of the documents which LSI tries to approximate and, the basis for clustering algorithm is the similarity of document which also depends on the document representation.

By incorporating background knowledge directly into the document representation via word weighting, reflecting similarity between the documents, we enable our methods to discover concepts which resemble the view that the user has on the data.

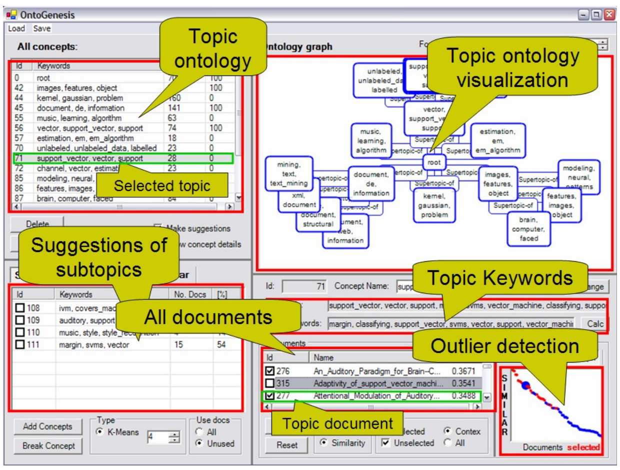

Figure 1. Screen shot of the interactive system for construction topic ontologies.

3 WORD WEIGHTING

3.1 Bag-of-Words and Cosine Similarity

Most commonly used representation of the documents in text mining is bag-of-words representation. Let $V = w_1, \ldots, w_n$ be vocabulary of words. Let TF_k be the number of occurrences of the word w_k in the document. In the bag-of-words representation a single document is encoded as a vector x with elements corresponding to the words from a vocabulary, eg. $x^k = TF_k$. These vectors are in general very sparse since the number of different words that appear in the whole collection is usually much larger than the number of different words that appear inside one specific document.

Measure usually used to compare text documents is the cosine similarity and is defined to be the cosine of the angle between two documents' bag-of-words vectors,

$$sim(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{k=1}^{n} \mathbf{x}_i^k \mathbf{x}_j^k}{\sqrt{\sum_{k=1}^{n} \mathbf{x}_i^k \mathbf{x}_i^k} \sqrt{\sum_{k=1}^{n} \mathbf{x}_i^k \mathbf{x}_i^k}}. \quad (1)$$

Performance of both bag-of-words representation and cosine similarity can be significantly improved by introducing word weights. Each word from vocabulary V is assigned a weight and elements of vectors $\mathbf{x}_i$ are multiplied by the corresponding weights.

As we already mentioned, our approach is based on the word weights being the key to viewing the same data from different angels. We can use the weights to store the background knowledge since the weights define which words are important.

3.2 TFIDF

Most of the research on word weighting schemas was traditionally done in the information retrieval community. A typical goal in information retrieval is to find the most relevant document from the document collection for a given query. Many popular methods from information retrieval are based on measuring cosine similarity between the documents and a query and their performance can be significantly improved by appropriate weighting of the words.

Most of the popular methods for this task developed in last decades do not involve learning. Word weights are calculated by predefined formulas from some basic statistics of the word frequencies inside the document and inside the whole document collection [10]. These methods are base on intuition and experimental validation.

The most widely used is the TFIDF weighting schema [10] which defines elements of bag-of-words vectors with the following formula:

$$\mathbf{x}_i^k = TF_k \cdot \log(N \cdot IDF_k). \quad (2)$$

The intuition behind this weighting schema is that the words which occur very often are not so important for determining if a pair of documents is similar while a not so frequent words occurring in the both documents is a strong sign of similarity. The TFIDF weighting can be easily modified to include category information by replacing IDF and number of documents with ICF and number of categories.

There are many extensions of this schema most famous being Okapi weighting schema [9] which we will skip here since it does not incorporate category information.

3.3 SVM Feature Selection

As we will see in the next chapter a different approach can also be taken for generating word weights based on feature selection methods. Feature selection methods based on Support Vector Machine (SVM) [2] has been found to increase the performance of classification by discovering which words are important for determining the correct category of a document [1].

The method proceeds as follow. First linear SVM classifier is trained using all the features. Classification of a document is done by multiplying the document's bag-of-words vector with the normal vector computed by SVM,

$$x^T w = x^1 w^1 + x^2 w^2 + \ldots + x^n w^n, \quad (3)$$

and if the result is above some threshold b then the document is considered positive. This process can also be seen as voting where each word is assigned a vote weight w^i and when document is being classified each word from the document issues $x^i w^i$ as its vote. All the votes are summed together to obtain the classification. A vote can be positive (document should belong to the category) or negative (the document should not belong to the category).

A simple and naive way of selecting the most important words for the given category would be to select the words with the highest vote values wi for the category. It turns out that it is more stable to select the words with the highest vote $x^i w^i$ averaged over all the positive documents.

The votes w^i could also be interpreted as word weights since they are higher for the words which better separate the documents according to the given categories.

3.4 Word Weighting with SVM

The algorithm we developed for assigning weights using SVM feature selection method is the following:

1. Calculate a classifier for each category from the document collection (one-vs-all method for multi-class classification). TFIDF weighting schema can be used at this stage. Result is a set of SVM normal vectors $W = \{\mathbf{w}_j; j = 1, \ldots, m\}$, one for each category.
2. Calculate weighting for each of the categories from its classifier weight vector. Weights are calculated by averaging votes $x^i w^i$ across all the documents from the category. Only weights with positive average are kept while the negative ones are set to zero. This results in a separate set of word weights for each category. By μ_k^j we denote weight for the k-th word and j-th category.
3. Weighted bag-of-words vectors are calculated for each document. Let $C(d_i)$ be a set of categories of a document d_i. Elements of vector $\mathbf{x}_i$ are calculated in the following way:

$$\mathbf{x}_i^k = \left(\sum_{j \in C(d_i)} \mu_k^j \right) \cdot TF_k \ . \quad (4)$$

This approach has another strong point. Weights are not only selected so that similarities correspond to the categories given by the user but they also depend on the context. Let us illustrate this on a sample document which contains words "machine learning". If the document would belong to category "learning" then the word "learning" would have high weight and the word "machine" low weight. However, if the same document would belong to category "machine learning", then most probably both words would be found important by SVM.

4 PRELIMINARY RESULTS

4.1 Reuters RCV1 Dataset

As a document collection for testing our method we chose Reuters RCV1 [7] dataset. The reason for which we chose it is that each news article from the dataset has two different types of labels (categories). Each news article is assigned labels according to (1) the topics covered and (2) the countries involved in it. We used a subset of 5000 randomly chosen documents for the experiments.

A List with the 10 most frequent categories from the used subset of RCV1 dataset is shown in Table 1. The statistics are for the subset used in the experiments.

Table 1. List of 10 most frequent categories for topics and countries view.

Topics view			Countries view	
CCAT	corporate/industrial	46%	USA	33%
GCAT	government/social	30%	UK	11%
MCAT	markets	24%	Japan	6%
C15	performance	19%	Germany	4%
ECAT	economics	14%	France	4%
C151	accounts/earnings	10%	Australia	3%
M14	commodity/markets	10%	India	3%
C152	comment/forcast	9%	China	3%
GPOL	domestic politics	7%	EEC	3%
M13	money markets	7%	Hong Kong	2%

4.2 Results

In the Figure 2 are the top 3 concepts discovered with k-means algorithm for both word weighting schemas. Documents are placed also in different concepts. For example, having two documents talking about the stock prices, one at the New York stock-exchange and the other at the UK stock-exchange. The New York document was placed in (1) Market concept (the same as the UK document) and in (2) USA concept (while the UK document was placed in (2) Europe concept).

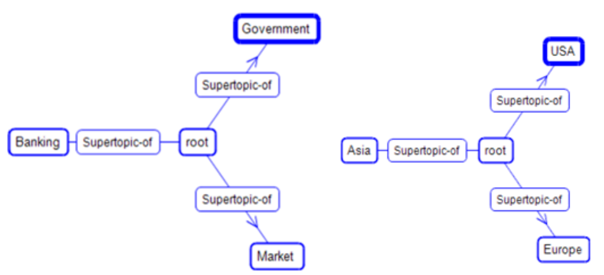

Figure 2. The top 3 discovered concepts for topic labels (left) and for country labels (right).

5 CONCLUSION

In this paper we have presented a method for learning document similarity measure trough selecting appropriate word weights for bag-of-words document representation model. We selected the word weights by training the SVM linear classifier for given categories and than extracting the word weights from the hyper plane normal vector. The learned word weighting schema was used to adjust the concept discovery methods in the OntoGen system to the user's domain knowledge.

As part of the future work we plan to extend this method to the text categorization task where category information is known only for the documents from training set.

ACKNOWLEDGEMENTS

This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under SEKT Semantically Enabled Knowledge Technologies (IST-1-506826-IP), NeOn Networked Ontologies (IST-2004- 27595) and PASCAL Network of Excellence (IST-2002-506778). This publication only reflects the authors' views.

REFERENCES

[1] Brank, J., Grobelnik, M., Milic-Frayling, N., Mladenic, D.: Feature selection using support vector machines. Proceedings of the Third International Conference on Data Mining Methods and Databases for Engineering, Finance, and Other Fields, Bologna, Italy, 25–27 September 2002

[2] N. Cristianini and J. Shawe-Taylor, An introduction to support vector machines, Cambridge University Press, 2000

[3] S. Deerwester, S. Dumais, G. Furnas, T. Landuer and R. Harshman, Indexing by Latent Semantic Analysis, Journal of the American Society of Information Science, vol. 41, no. 6, 391-407, 1990

[4] Fortuna, B., Mladenic, D., Grobelnik, M., (2005a). Semi-automatic construction of topic ontology. Proceedings of the ECML/PKDD Workshop on Knowledge Discovery for Ontologies.

[5] Grobelnik M. & Mladenic D. Automated knowledge discovery in advanced knowledge management. J. of. Knowledge management 2005, Vol. 9, 132-149.

[6] Jain, Murty and Flynn: Data Clustering: A Review, ACM Comp. Surv., 1999

[7] Lewis, D. D., Yang, Y., Rose, T., and Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361-397, 2004.

[8] Novak B., Mladenic D. & Grobelnik M. Text classification with active learning. Proceedings of GfKl 2005.

[9] Robertson, S. E., S. Walker, M. M. Hancock-Beaulieu, M. Gatford and A. Payne. Okapi at TREC-4. The Fourth Text REtrieval Conference (TREC-4). 1996

[10] G.Salton. Developments in Automatic Text Retrieval. Science, Vol 253, pages 974-979. 1991

[11] Singhal, A., C. Buckley and M. Mitra. Pivoted Document Length Normalization. Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval. 1996

Online Ontological Reasoning for Context-Aware Internet Services [1]

Alessandra Agostini and **Claudio Bettini** and **Daniele Riboni**[2]

The research group of the DaKWE laboratory at the University of Milan has been working for the last three years at the specification and implementation of a middleware – named *CARE*[3] – to support context-aware service adaptation for mobile users. *CARE* has three major goals: a) supporting the fusion and reconciliation of context data obtained from distributed sources, b) supporting context dynamics through an efficient form of reasoning, and c) capturing complex context data that go beyond simple attribute-value pairs.

While goal b) has been considered in other works [6, 11], it becomes more difficult to achieve when different sets of inference rules are provided by distributed sources. Even more difficult is to conciliate efficient reasoning with the expressiveness requirements imposed by the goal c).

The *CARE* middleware and its underlying technical solutions have been presented in [1, 3]. In our framework the contextual data, being by nature distributed, is managed by different entities (i.e., the user, the network operator, and the service provider). We call *profile* a subset of context data collected and managed by a certain entity. Each entity has a dedicated Profile Manager for handling its own context data. Profiles include both shallow context data and ontology-based context data which is expressed by means of references to ontological classes and relations. Both the user and the service provider can declare policies in the form of rules over profile data which guide the adaptation and final personalization of the service. A dedicated module is in charge of building the aggregated context data for the application logic. In particular, it evaluates adaptation policies and solves possible conflicts arising among context data and/or policies provided by different entities. The ad-hoc rule-based reasoner is particularly efficient if no ontological reasoning is performed, having linear complexity. Experimental results have shown that the evaluation of rules is executed in few milliseconds.

In our framework we need to model both simple context data such as device capabilities or current network bearer, and socio-cultural context data describing, for instance, the user current activity, the set of persons and objects a user can interact with, and the user interests. While the first category, that we call *shallow* context data, can be naturally modeled by means of attribute/value pairs, the second one calls for more sophisticated representation formalisms – such as ontologies – and we call it *ontology-based* context data. Similarly to other research works (e.g., [5] and [7]), we have adopted OWL [10] as the language for representing ontology-based context data. This choice is motivated by the fact that the description logic languages underlying the *Lite* and *DL* sublanguages of OWL guarantee completeness and decidability, while promising high expressiveness. Moreover, a number of tools already exist for processing OWL ontologies and, being OWL a W3C Recommendation, the available utilities should further increase.

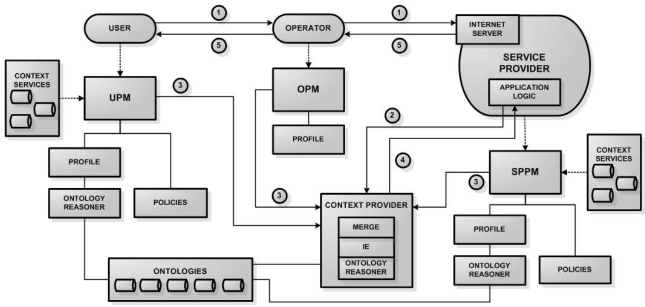

Figure 1. The *CARE* middleware architecture.

For a framework in which efficiency is a fundamental requirement, the introduction of ontological reasoning is particularly challenging. The hybrid approach implemented in *CARE* is based on a loose interaction between ontological and rule-based reasoning. While rule-based reasoning is performed at the time of the service request, ontological reasoning is mostly performed asynchronously by profile managers. However, in particular cases, ontological reasoning must be performed at the time of the user request, after having populated the ontology with instances collected from the distributed profile managers. In order to illustrate the hybrid mechanism, suppose that a user declared a policy rule asking to set her status to *busy* when involved in a business meeting:

$$\textbf{If}\ \ \textit{Activity} = \textit{`BusinessMeeting'}\ \ \textbf{then}\ \ \textit{Status} = \textit{`Busy'} \qquad (1)$$

Since the rule precondition predicate *Activity* is an ontology-based context parameter, its value must be inferred through ontological reasoning before evaluating the rule.

As an example, consider a possible definition of the *BusinessMeeting* activity:

$$\begin{aligned} \textit{BusinessMeeting}\ \equiv\ & \textit{Activity} \sqcap\ \geq 2\,\textit{Actor}\ \sqcap \\ & \forall\,\textit{Actor.Employee}\ \sqcap\ \exists\,\textit{Location.WorkLocation} \end{aligned}$$

[1] This work has been partially supported by Italian MIUR (FIRB "Web-Minds" project N. RBNE01WEJT_005).
[2] DICo, University of Milan, via Comelico 39, I-20135 Milan, Italy, email: {agostini,bettini,riboni}@dico.unimi.it
[3] Context Aggregation and REasoning middleware.

Based on this definition, in order to check whether the user is involved in a business meeting it is necessary to have information about the people she is with (possibly derived by the user profile manager analyzing her agenda) and her current location (possibly provided by the network operator). This data is added to the assertional part of the ontology (i.e., the *ABox*).

Our initial experimental setup was based on the realization of the whole ABox upon receiving the context data from the profile managers. The current user activity was identified by performing nRQL queries to the well-known description logic reasoner Racer [8].

Even if OWL-DL guarantees completeness and decidability, performing online reasoning tasks with an OWL ontology could be computationally unfeasible, especially when providing an interactive service to a possibly huge number of users. Despite several assessments on the performance of reasoning with description logics are available, we performed some tests in order to verify the feasibility of executing ontological reasoning at the time of the service request with our specific OWL-DL ontologies. As expected, experimental results showed that query response times are strongly correlated to the number of instances of the examined ontology class as well as to the depth of the class within the ontology hierarchy. Our results confirmed that the execution of these ontological reasoning tasks at the time of the service request is unfeasible, even having a small ontology populated with few instances. In particular, query response times in our experiments are in the order of seconds.

We are investigating alternative approaches for overcoming the above mentioned computational issues. A possible solution consists in keeping the terminological part of the ontology (i.e., the *TBox*) static, in order to be able to perform the TBox classification [2] offline. In this way it is possible to save a good amount of computational time while serving user requests, since the ontology classification task is particularly expensive.

Furthermore, the assertional part of the ontology can be filled offline with those instances that are known *a priori*, i.e., before retrieving context data from the distributed profile managers. This data obviously depends on the particular domain addressed by the ontology. In the case addressed by our example, the ABox should be populated with a huge number of instances, including those that correspond to the employees of the user organization, and to particular locations (e.g., rooms belonging to the organization). After having populated the ontology with these instances, it is possible to perform the ABox realization [2] offline. Once again, ABox realization is an expensive reasoning task, which is unsuitable to perform online when the ontology contains a huge number of instances.

At the time of the user request, the ABox is filled with only those instances that are retrieved from the profile managers. Considering the ontology definition (1) of our example, the instances to be inserted into the ontology correspond to a new activity *currentActivity* – the one performed by the user – and to the relations that link that activity to its actors and location. Adopting this approach, the only reasoning task that must be performed online is the *instance checking* of the single *currentActivity* instance with respect to the *BusinessMeeting* concept.

As a preliminary step for assessing the feasibility of this approach, we are going to perform extensive experiments for estimating the execution times of this task in relation to various dimensions, including the TBox size, the number of instances that are known *a priori*, and the number of instances that are introduced into the ABox at the time of the user request.

Moreover, we are interested in testing some optimization techniques aimed at improving the efficiency of ABox reasoning. These optimizations are based on the use of relational database techniques. A well-known proposal in this sense is the InstanceStore system [9]. However, at the time of writing, InstanceStore has some limitations that are critical for our reasoning scenarios. Indeed, it does not allow the introduction of relations between individuals into the ABox. An alternative proposal for optimizing ABox reasoning by means of DBMS techniques can be found in [4]. Since in this case relations between individuals are supported, we are investigating the use of similar techniques in our framework.

REFERENCES

[1] A. Agostini, C. Bettini, N. Cesa-Bianchi, D. Maggiorini, D. Riboni, M. Ruberl, C. Sala, and D. Vitali, 'Towards Highly Adaptive Services for Mobile Computing', in *Proceedings of IFIP TC8 Working Conference on Mobile Information Systems (MOBIS)*, pp. 121–134. Springer, (2004).

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

[3] C. Bettini and D. Riboni, 'Profile Aggregation and Policy Evaluation for Adaptive Internet Services', in *Proceedings of The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous)*, pp. 290–298. IEEE Computer Society, (2004).

[4] Cui Ming Chen, 'Large Abox Store (LAS): Database Support for Abox Queries', Master Thesis, Concordia University, Montreal, Quebec, Canada, (September 2005).

[5] Harry Chen, Filip Perich, Timothy W. Finin, and Anupam Joshi, 'SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications.', in *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), Networking and Services*, pp. 258–267. IEEE Computer Society, (2004).

[6] B. Grosof, 'Prioritized Conflict Handling for Logic Programs', in *Proceedings of the International Logic Programming Symposium (ILPS)*, pp. 197–211, (1997).

[7] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, 'An Ontology-based Context Model in Intelligent Environments', in *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, (2004).

[8] Volker Haarslev and Ralf Möller, 'RACER System Description', in *Proceedings of Automated Reasoning, First International Joint Conference (IJCAR 2001)*, pp. 701–706. Springer, (2001).

[9] Ian Horrocks, Lei Li, Daniele Turi, and Sean Bechhofer, 'The Instance Store: DL Reasoning with Large Numbers of Individuals', in *Proceedings of the 2004 International Workshop on Description Logics (DL2004)*, volume 104 of *CEUR Workshop Proceedings*. CEUR-WS.org, (2004).

[10] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen, 'From SHIQ and RDF to OWL: The Making of a Web Ontology Language', *Journal of Web Semantics*, **1**(1), 7–26, (2003).

[11] R. Hull, B. Kumar, D. Lieuwen, P. Patel-Schneider, A. Sahuguet, S. Varadarajan, and A. Vyas, 'Enabling Context-Aware and privacy-Conscius User Data Sharing', in *Proceedings of the 2004 IEEE International Conference on Mobile Data Management*, pp. 187–198. IEEE Computer Society, (2004).

Reasoning with temporal context in news analysis

Igor Mozetič and **Damjan Bojadžiev**[1]

Abstract. One aspect of implicit, contextual information is its temporal component. Explicating this component in a formal model makes it possible to disambiguate some context-dependent expressions and discover connections between expressions. We have implemented and extended Allen's algebra of temporal intervals in a reasoner that takes into account the linear nature of time and the granularity of temporal expressions (days/weeks/...). If this algebra is used to model the temporal extension of events, the reasoner can track and connect the reference of indexical expressions about them. We intend to use the reasoner for analysing news streams, to help discover connections between news items.

1 INTRODUCTION

This paper expresses the combination of our interests in the subjects of context and ontologies, taken by themselves and in their connections. On the more abstract, logico-philosophical side, there are the questions of definition and significance: what *is* context, what is specific about it, and how does it, its inclusion or its omission, affect cognitive, deductive and computational processes. For example, getting stuck into loops, for humans and for machines, might be conceived as a loss of context. Judging by the many definitions of context in different disciplines, the notion of context is itself context-sensitive, and it is hard to point out the specific characteristic that distinguishes context from background, prior knowledge and/or the multiplicity of implicit facts and assumptions that is simply taken for granted, unnoticed, left out or suppressed as too obvious to mention. This is reflected in the reluctance in some important papers on context to actually define it, such as McCarthy's [8], and in his insistence that "there is no universal context". In connection with ontologies, there is also some context-dependence in the definition of context: ontologies supply context for browsing [5] (which again indicates that context can be practically anything), but mappings between ontologies supply context too, as in C-OWL [6].

On the more computational side, our interest is in ontologies of time, or of the ways we refer to its passage, and in actual implementations of automatic reasoning about temporal information. We have implemented Allen's axiomatization of temporal relations, used eg. in DAML-Time and SUMO [10], in the constraint logic programming system CLP($\mathcal{Q}$) [9]. We plan to use this implementation in automatic news (stream) analysis, for disambiguating context-dependent reference and for news classification. The remainder of this paper gives more information about the axiomatization and its implementation, and some examples of the intended application. More generally, we have a hunch that some of the work done at our Department, eg. on user profiling and on simultaneous ontologies [7], can be formulated as programming context dependency, and we are working on a convincing formulation.

[1] Jozef Stefan Institute, Ljubljana, Slovenia, email: igor.mozetic@ijs.si

In our news analysis system, we are mainly concerned with the temporal aspects of context. The system will take into account the time-stamp part of the metadata about news items, and temporal models of the events reported, to distinguish related news items from unrelated ones. Thus, our working hypothesis about temporal context could be expressed in the equation

$$\boxed{\text{temporal context(news)} = \text{temporal model} + \text{metadata(news)}}$$

This definition was originally inspired by [11], which deals with contextual vocabulary acquisition (how to infer the meaning of a new word from textual clues). It identifies two components of context: prior knowledge (which is subject to belief revision) and co-text of the word to be learned. In our case, the task is to find a semantic link between news items. The context of news is prior knowledge in the form of a temporal model, and the metadata that comes with the news. We do not deal with the model revision component, and restrict our system to temporal aspects. However, causal, spatial and other types of models and/or ontologies also represent prior knowledge and thus fit into our definition of context. If the restrictions to temporality and subject matter (news items) are dropped, the equation above generalizes to the form

$$\boxed{\text{context(X)} = \text{prior knowledge} + \text{co-data(X)}}$$

2 TEMPORAL ALGEBRA AND ONTOLOGY

Allen [4] proposed an interval algebra to represent relative temporal information, such as the order of events. The representation of events by time intervals rather than points allows the expression of hierarchical, indefinite and incomplete information, at different levels of granularity. The temporal algebra uses the thirteen possible relations between time intervals, such as one interval starting or finishing another interval, or being before or meeting another one.

To represent indefinite and incomplete information, Allen uses disjunction to allow any subset of the basic relations to hold between two time intervals. A set of temporally related events forms a network, with edges corresponding to (possibly disjunctive) relations between events. There are two fundamental queries one can pose about such a network:

- Find the feasible relations between all pairs of events, and
- Determine the consistency of the temporal relations.

When we came accross this algebra, we were not aware of any (complete) reasoner for it. Since its networks of temporal relations express constraints on relations between intervals, we decided to implement the algebra in a constraint logic programming system CLP($\mathcal{Q}$) [9]. The implementation allows automatic reasoning about temporal events, such as:

- "If X precedes Y, and Y overlaps with Z, what are the possible temporal relations between X and Z ?"
- "If X takes longer then Y, can X occur during Y ?"
- "Given a set of temporally related events, what are the possible consistent scenarios on the time line ?"

On top of this basic implementation, we formulated a generic ontology of time which covers everyday concepts such as hours, days, seasons, and the relations between them. Note that there is no fixed underlying time scale. This time ontology is similar to the specifications in SUMO [3], DOLCE [2], and DAML-Time [1], and has the advantage of being executable.

3 NEWS ANALYSIS

Let us first illustrate the desired feature of the analysis system by a simple example. Suppose we receive two news items on two subsequent days:

- **Day1**: "Giant waves hit the shore early today."
- **Day2**: "An ocean floor earthquake was detected yesterday."

One interesting question that a news analyst might then ask is: Are these news items related?

There are various techniques used for news analysis, but all essentially measure the degree of similarity between items. The metrics used can be purely syntactic or increasingly based on semantics. We might roughly distinguish three levels of (semantic) similarity:

1. purely lexical, based only on the presence of keywords
2. weak or lexicographic, taking into account taxonomic meaning
3. strong, using models of word referents

The models in question are formulated in terms of the temporal ontology; in the case of the news items above, a relevant example would be the temporal model of a tsunami, shown in Figure 1.

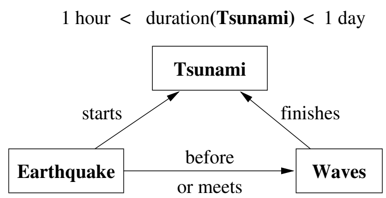

Figure 1. A simple temporal model of a tsunami.

To detect whether the news items above are related or not, we would use the following algorithm:

1) When the first news (waves) arrive, find the temporal terms ("today") and resolve them locally, with respect to the news metadata (time-stamp). The implicit temporal reference can then be transformed into explicit reference in terms of the temporal ontology, resulting in the temporal relation

Waves during **Day1**

2) When the next news arrive (earthquake), the procedure gives

Earthquake during previous(**Day2**)

Here, the reference "yesterday" is expressed by applying the function "previous" to the current **Day2**.

3) Reasoning with the temporal ontology gives (Figure 2):

previous(**Day2**) equals **Day1**

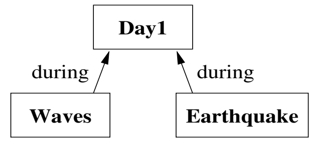

Figure 2. Temporal relations between both news events.

4) Reasoning with the tsunami model then shows that the news are consistent with a tsunami. Therefore, we can formulate a defeasible hypothesis: A tsunami is a possible explanation of the two events, which links the news items in question.

5) However, if the news say that

Waves before **Earthquake**

the tsunami link will be ruled out as a possible explanation of the news sequence.

In this way, the temporal model can provide a stronger measure of semantic similarity and thus increase the quality of the news analysis system.

4 CONCLUSION

Our definition of temporal context seems useful, especially for a news analysis system, because it encompasses both static prior knowledge and dynamic metadata (a sophisticated example of the use of such data in reasoning with abductive constraint logic programs is presented in [12]). If experiments with the news analysis system, augmented by the temporal ontology, the constraint logic program and temporal models such as the tsunami model, prove successful, other semantic models, such as causal and spatial, will be included too. In the tsunami example, these would be needed to capture other relevant relations, such as the fact that the earthquake needs to take place under the see in roughly the same geographic area.

ACKNOWLEDGEMENTS

Supported by the Slovenian Research Agency, and the EU Projects SEKT (IST-2003-506826) and NeOn (IST-2004-27595).

REFERENCES

[1] Daml-time. http://www.cs.rochester.edu/ferguson/daml/.
[2] Dolce. http://www.loa-cnr.it/DOLCE.html.
[3] Sumo. http://www.ontologyportal.org/.
[4] J.F. Allen, 'Maintaining knowledge about temporal intervals', *Communications of the ACM*, **26**, 832–843, (1983).
[5] N. Aussenac-Gilles and J. Mothe, 'Ontologies as background knowledge to explore document collections', in *Proc. RIAO-2004*, pp. 129–142, Avignon, (2004).
[6] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt, 'C-owl: Contextualizing ontologies', in *Proc. 2nd ISWC*, pp. 164–179, (2003).
[7] Grobelnik M. Fortuna, B. and D. Mladenič, 'Background knowledge for ontology construction', in *Proc. Contexts and Ontologies*, (2006).
[8] J. McCarthy, 'Notes on formalizing context', in *Proc. 13th IJCAI*, (1993). http://www-formal.stanford.edu/jmc/context3/context3.html.
[9] I. Mozetič, 'Temporal interval reasoning with clp(q)', in *Proc. 8th IS-2005*, pp. 178–181, Ljubljana, (2005).
[10] I. Niles and A. Pease, 'Towards a standard upper ontology', in *Proc. 2nd FOIS-2001*, Ogunquit, Maine, (2001).
[11] W.J. Rapaport, 'What is the 'context' for contextual vocabulary acquisition?', in *Proc. 4th ICCS/ASCS-2003*, pp. 547–552, Sydney.
[12] Madnick S. Zhu, Hongwei and M. Siegel, 'Reasoning about temporal context using ontology and abductive constraint logic programming', Technical Report 15, CISL, MIT, Cambridge, (2004).

Ontologies and Context for Educational Process Modeling in IMS Learning Design

Colin Knight[1] and **Dragan Gašević**[2] and **Griff Richards**[3]

Abstract. This paper discusses the role of context and ontologies in educational process modeling. Educational process modeling seeks to represent the complex interactions that take place in multi-actor learning environments, with the view that the sequence and types of interactions can be equally as important as the sequence and types of content. The IMS Learning Design specification provides the semantics to represent multi-actor interactions within an educational process, and the IDLD project has resulted in a substantial catalogue of Learning Design models from a variety of contexts. To facilitate reuse of these models in different contexts, ontologies have been developed based on the IMS Learning Design specification and we propose to use context to determine the relevance of Learning Design models when used in new situations to guide the learning process.

1 INTRODUCTION

Learning design has emerged at the forefront of research into the modeling of dynamic, learner-centered eLearning experiences. The IMS Learning Design (LD) Specification [1] provides a way to represent complex multi-actor interactions in an educational environment. This specification has been widely integrated into a number of learning management systems and authoring tools, and several ontologies have been developed around the specification [2],[3],[4]. The recently-completed IDLD [5] project effort involved modeling the educational processes of dozens of actual on-line and face-to-face courses being delivered at universities across Canada according to the IMS LD specification. Effective reuse of Learning Design models remains a challenge because each model is gathered from a diverse learning situation, meaning that some of elements of the model become irrelevant when applied to new situations. Since it is a time-consuming task to model the educational process in a learning design, a mechanism is needed to guide the transfer of models for use in new situations. We propose the use of contexts as a solution to this problem.

Previous work on development of ontologies for eLearning has focused on the authoring process [6],[7] and sequencing of content [8],[9] with relatively little emphasis on expressing the role of multi-actor interactions in the learning process. However, these efforts have provided a useful framework for us to work within.

An exploration of contextual variables for learning environments has been completed in [2]. Since the variability of these environments is almost as great as the variability of the designs themselves, it is necessary to simplify to include the context elements that had the greatest influence on the structure and sequence of the course structure. This simplification usually occurs after consultation with the course author or instructor.

2 LOCO - an ontology compatible with IMS-LD

The IMS-LD Information Model and XML binding is the specification for Learning Design [1]. The LOCO ontology [2] is a light-weight ontology in the OWL language, based on the IMS-LD Information Model. The ontology is able to represent complex series and parallel interactions of actors. Each actor is assigned one or many *Roles*, which are associated with *Activities* according to the *Role-part* that each actor engages in and in *Environment* (resources, services) in which the activities take place.

[1] Simon Fraser University, Canada. Email cjk2@sfu.ca

[2] Simon Fraser University, Canada. Email: dgasevic@sfu.ca

[3] Simon Fraser University, Canada. Email: griff@sfu.ca

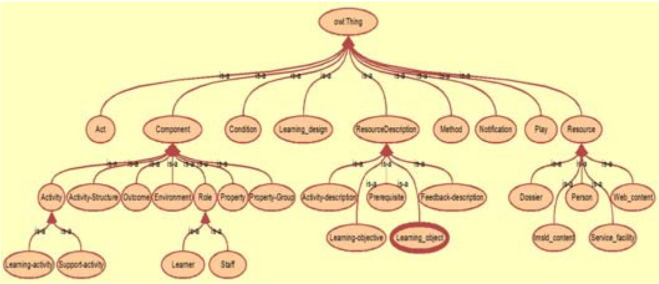

Figure 1. The class hierarchy of the LOCO ontology

To create the LOCO, some changes were made to the Information Model [1] in order to conform to established best-practice recommendations for ontology design [11], and to resolve some ambiguities and inconsistencies in the information model. These changes are described in detail in [2]. To date the LOCO only addresses IMS-LD Level A.

3 CONCLUSIONS

Treatment of context information stands as a barrier to the reusability of Learning Design. Existing solutions for the use of context and ontologies in learning applications could be enhanced by incorporating educational process modeling into the semantic representation of the learning space. A method of using context to effectively transfer these processes to new settings would greatly benefit learners by enabling pedagogical methods such as collaborative learning to become more of reality in eLearninig. Also, reusability would enhance the cost-effectiveness of modeling learning designs. Future research will involve determining suitable methods for using context effectively to transfer learning designs to new learning situations with minimal redesign effort.

REFERENCES

[1] IMS Global Learning Consortium (2003). IMS Learning Design Information Model. Version 1.0 Final Specification, revision 20. Retrieved March 25, 2005, from http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html.

[2] Knight, C. Gasevic, D. & Richards, G. (2005). Ontologies to integrate learning design and learning content. *Journal of Interactive Media in Education*. Special Issue on Advances in Learning Design, 2005 (7).

[3] Psyche, V., Bourdeau, J., Nkambou, R., Mizoguchi, R. (2005). Making Learning Design Standards Work with an Ontology of Educational Theories. In proceedings of the 12th Annual Conference on Artificial Intelligence in Education, Amsterdam, NL.

[4] Amorim, R., Lama, M., Sanchez, E., Riera, A., Vila, X. (2006). A Learning Design Ontology Based on IMS-LD. Jounal of Education, Technology, and Society, 9 (1), 38-57.

[5] Paquette, G., Marino, O., Lundgren-Cayrol, K., Léonard, M., I de la Teja , I. (2006) The IDLD Repository – Classification and Repurposing of Learning Designs. .TENCompetence Workshop, Sofia, Bulgaria, March 20.

[6] Mizoguchi, R., and Bourdeau, J. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems. *International Journal of Artificial Intelligence in Education*, 11, 107-121.

[7] Inaba, A., & Mizoguchi, R. (2004). Learning design palette: An ontology-aware authoring system for learning design. In proceedings of the International conference on computers in education, Melbourne, Australia.

[8] Jovanović, J., Gašević, D., Verbert, K., Duval, D. (2005). Ontology of learning object content structure. *In Proceedings of the 12th International Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 322-329.

[9] Dichev, C., & Dicheva, D. (2005). Contexts as Abstractions of Grouping. In proceedings of the Context & Ontologies Workshop 2005, Pittsburg, Pennsylvania, USA.

[10] Mizoguchi, R., Vanwelkenhuysen, J., and Ikeda, M. (1995). Task Ontology for Reuse of Problem Solving Knowledge. Towards Very Large Knowledge Bases. Mars, N.J.I (eds), IOS Press.

[11] Noy, F. N. & McGuinness, D.L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report SMI-2001-0880, Stanford Medical Informatics,Stanford University, Stanford, CA, USA.