Modelling your Domain using Ontologies

Successes and Challenges

Thomas Meyer
Meraka Institute, CSIR
Pretoria, South Africa
tommie.meyer@meraka.org.za

ABSTRACT

This paper provides a brief overview of the use of ontologies in Computer Science and argues that one of the main reasons for its popularity in recent years is that it has become possible, in practice, to perform a variety of *reasoning tasks* over large ontologies. We also, briefly, consider some of the challenges to be overcome before it can be seen as having reached the status of an established technology.

1. INTRODUCTION

In the past fifteen years, advances in technology have ensured that access to vast amounts of data is no longer a significant problem. Paradoxically, this abundance of data has lead to a problem of information overload, making it increasingly difficult to locate *relevant* information.

An area in which this dilemma is illustrated particularly well is on the World Wide Web. For many of us, life without access to the Web is now virtually impossible to imagine. It provides access to amounts of information that would have been simply unthinkable ten to fifteen years ago. But the problem is that the current Web is, for the most part, aimed at humans who determine the relevance of a web page by manually perusing its content. What is needed is a way to automate the process of determining which pages contain relevant information. At present the best way to do so is through syntactic searches in which keywords are viewed as patterns of characters, and are matched to strings occurring in Web resources. Given these limitations it is illuminating to bear in mind that Tim Berners-Lee, the founder of the Web, originally had a much more ambitious vision. He envisaged a Web in which machines would have access to the *meaning* of the available information: a Semantic Web with information shared and processed both by automated tools, such as search engines, and by human users [6].

The basic approach to realising this vision is to ensure that information resources on the Semantic Web will not only contain data, but also metadata, describing what the data are about. This will allow machines and their human users to identify, collect and process suitable information sources by interpreting the semantic metadata based on the task at hand. The crucial part in all of this, the method for describing what the data are about, is provided by the use of *ontologies*. In Computer Science the term "ontology" refers to a designed artefact consisting of a specific shared vocabulary used to describe entities in some domain of interest, and a set of assumptions about the intended meaning of the terms in the vocabulary. In other words, an ontology structures information in ways that are appropriate for a specific application domain, and in doing so provides a way to attach meaning to the terms and relationships used in describing the domain. A more formal, and widely used, definition, is that of Grüber [13] who defines an ontology as a formal specification of a conceptualisation. The importance of this technology is evidenced by the growing use of ontologies in a variety of application areas, and is in line with the view of ontologies as the emerging technology driving the Semantic Web initiative.

In the rest of this overview we briefly consider some of the main successes in the field of ontologies, and also consider some of the challenges that still need to be overcome.

2. SUCCESSES

The vision of the Semantic Web is still far from being realised, but it forms part of a growing body of work on ontologies, applicable to a much wider variety of application domains. While the use of ontologies, in one form or another, is nothing new, advances in recent years have made it possible to apply this technology in ways that no one would have dreamt of a few years ago. The biggest successes have been associated with the advances made in *reasoning* over large ontologies, and as a result, there is growing interest in the use of ontologies and related semantic technologies in a wide variety of application domains.

Arguably the most successful application area in this regard is the biomedical field [32, 14]. Some of the biggest breakthroughs in ontological reasoning can be traced back to the pioneering work of Horrocks [15] who developed algorithms specifically tailored for medical applications. And recent advances have made it possible to perform reasoning tasks on large-scale medical ontologies that would have provoked disbelief ten years ago.

As an example, consider the case of SNOMED CT which is

being adopted as the standard medical ontology in a growing number of countries worldwide, including the United States, the United Kingdom, Australia, and a number of European countries.[1] It is a phenomenally large ontology with over 300 000 concepts and more than 1 300 000 relations between concepts. The correct classification of concepts—determining how the different concepts are related to another—is crucial to the practical utilisation of SNOMED CT. Using state-of-the-art reasoning techniques it is now possible to classify the whole of SNOMED CT in less than half an hour, a feat that would have been considered impossible ten years ago [30].

Apart from medical systems, ontologies have also been used in domains such as natural language processing, representing biological terminologies, configuration of technical systems, and databases. And as mentioned above, it now also forms the backbone of the Semantic Web. In these applications, the use of ontologies allows for the sharing of information between different agents. To make sure that different agents have a common understanding of the terms used, one needs ontologies in which these terms are described, and which thus establish a joint terminology between the agents. Thus, the construction, integration, and evolution of high-quality ontologies greatly depends on the availability of ontology languages equipped with a well-defined semantics and powerful reasoning tools.

The solution to this problem was found in an existing class of logics, called description logics or DLs, that provide for both, and are therefore ideal candidates for ontology languages [5, 2]. That much was already clear fifteen years ago, but at that time, there was a fundamental mismatch between the expressive power and the efficiency of reasoning that DL systems provided, and the expressivity and the large knowledge bases that ontologists needed. Through the basic research in DLs of the last fifteen years, this gap between the needs of ontologists and the systems that DL researchers provide has finally become narrow enough to build stable bridges. In fact, the web ontology language OWL, which was accorded the status of a World Wide Web Consortium (W3C) recommendation in 2004, and is therefore the official Semantic Web ontology language, is based on an expressive DL.[2]

2.1 Description Logics as ontology languages

Description logics (DLs) [3] are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. The name "description logics" is motivated by the fact that the important notions of the domain are described by concept *descriptions*, i.e., expressions that are built from atomic concepts (unary predicates) and atomic roles (binary predicates) using the concept and role constructors provided by the particular DL. DLs differ from their predecessors, such as semantic networks and frames, in that they are equipped with a formal, *logic*-based semantics, which can, for example, be given by a translation into first-order logic.

Knowledge representation systems based on description logics (DL systems) provide their users with various inference capabilities (like subsumption and instance checking) that allow them to deduce implicit knowledge from the explicitly represented knowledge. In order to ensure reasonable and predictable behavior of a DL system, these inference problems should at least be decidable, and preferably of low complexity. Consequently, the expressive power of the DL in question must be restricted in an appropriate way. If the imposed restrictions are too severe, however, then the important notions of the application domain can no longer be expressed. Investigating this trade-off between the expressivity of DLs and the complexity of their inference problems has been one of the most important issues in DL research.

The focus of this research has, however, changed in the last 15 years. In the beginning of the 1990s, DL researchers investigated the border between tractable and intractable DLs [10], and systems that employed so-called structural subsumption algorithms, which first normalise the concept descriptions, and then recursively compare the syntactic structure of the normalised descriptions, were still prevalent. It quickly turned out, however, that structural subsumption algorithms (which usually are tractable, i.e., have a polynomial run-time) can handle only very inexpressive languages, and that one cannot expect a DL of reasonable expressive power to have tractable inference problems [22]. For expressive DLs, tableaux-based inference procedures turned out to be quite useful [1]. After the first such tableaux-based subsumption algorithm was developed by Schmidt-Schauss and Smolka [27] for the DL $\mathcal{ALC}$, this approach was extended to various other DLs and also to other inference problems.

The tableaux-based algorithms employed for DLs usually have a rather high worst-case complexity. The first DL systems employing such algorithms (like KRIS and Crack) were based on PSPACE-complete logics (like $\mathcal{ALC}$), and modern tableau-based DL reasoners [20] such as FaCT++, and RACER are based on very expressive DLs (like $\mathcal{SHIQ}$ or $\mathcal{SHOQ}(D)$) which have an ExpTime-complete subsumption problem. Despite the high worst-case complexity of the underlying logics and of the algorithms themselves, the systems actually behave very well in realistic applications. This is probably due to the fact that their implementors have developed a great variety of sophisticated optimisation techniques for tableau-based algorithms [16]. The optimised algorithms have been evaluated both on realistic knowledge bases (coming from applications) and on randomly generated knowledge bases. Nevertheless, the reason why they behave so well in practice is not yet clear.

Another important development in the early 1990s was the realisation that DLs are very closely related to propositional modal logics (MLs). For instance, the DL $\mathcal{ALC}$ turned out to be just a syntactic variant of the basic multi-modal logic K [25]. This allowed DL researchers to transfer many decidability and complexity results from modal logics to description logics. However, on the practical side, the transfer went mostly in the other direction. In fact, the provers for propositional modal logics developed in the ML community were not competitive with the best DL systems. Instead of viewing DLs as syntactic variants of propositional MLs, one can also view them as decidable fragments of first-order logic (FOL). Most of the DLs considered in the literature

[1] http://www.ihtsdo.org/
[2] http://www.w3.org/2004/OWL/

belong to known decidable fragments of FOL (such as the guarded fragment or the two-variable fragment with or without counting quantifiers) [25].

3. CHALLENGES

Despite the progress made in recent years, a number of obstacles still remain before the use of ontologies can be regarded as having reached the status of an established technology. Roughly speaking, these can be categorised into issues relating to *conceptual modeling* and *data usage*.

3.1 Conceptual modeling

One of the first observations in this regard is that there are currently no firmly established methodologies for ontology engineering. It is generally recognised that this is a research topic that warrants urgent attention [12]. Furthermore, although a variety of tools exist for ontology construction and maintenance [29, 17, 24], they remain accessible mainly to those with specialised knowledge about the theory of ontologies. One of the suggestions for dealing with this problem is to design ontology languages that are as close to natural language as possible, while still retaining the unambiguous semantics of a formal language [28]. A related approach is to use unstructured text to automatically identify concepts and relationships in application domains, and in doing so contribute to the semi-automated construction of ontologies [7].

A second major obstacle is that, while most tools for ontology construction and maintenance assume a static ontology, the reality is that ontologies are dynamic entities, continually changing over time for a variety of reasons. This has long been identified as a problem, and *ontology dynamics* is currently seen as an important research topic [26, 4, 19, 18].

3.2 Data usage

Once an ontology engineer is satisfied that the ontology indeed correctly represents an application domain, the role of reasoning changes, and shifts to the *inference* or *extraction* of new facts about the domain that are implicitly represented.

Assuming that the problems relating to conceptual modeling have been solved, and that it is possible to construct and maintain high-quaity ontologies, a number of stumbling blocks related to data usage still remain. The main problem is that most available data are currently in the form of unstructured or semi-structured text, or can be found in traditional relational database systems. The rich conceptual structures provided by ontologies are therefore of little use unless ways can be found to automate, or semi-automate, the process of populating ontologies with this data. Regarding data in textual form, there have been some recent attempts to perform semi-automated instantiation of ontologies from text [31, 7]. With regards to the data found in database systems, it is necessary to employ *data coupling*—finding ways of linking the data residing in database systems to the ontologies placed on top of such systems [23, 8].

Finally, once an ontology is populated, it becomes possible to use it as a sophisticated data repository to which complex queries can be posed, at least in principle. In practice, at least two challenges remain. The first is to perform query answering efficiently, a topics of ongoing research [9]. The second is to go beyond purely deductive reasoning to answer queries and to be more proactive. A good example of this type of reasoning occurs is medical diagnosis which is an instance of a form of reasoning technically known as *abduction* [11].

4. CONCLUSION

As illustrated in this overview, research into ontologies and description logics has a strong and healthy tradition of combining strong theoretical results with highly applied work. At the Knowledge Systems Group in the Meraka Institute we are currently investigating the use of recent breakthroughs in ontology research as a means of providing improved data quality in application domains that are of crucial importance in the South African context. At present we are focusing on the development of modelling and reasoning tools for the bio-medical domain. One of the major benefits of such tools is their long-term applicability to a wider range of application domains. Perhaps they may be of use in your domain as well.

5. REFERENCES

[1] Franz Baader and Ulrike Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69:5–40, 2001.

[2] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics for the semantic web. *KI – Künstliche Intelligenz*, 4, 2002.

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

[4] Franz Baader, Carsten Lutz, Maja Milicić, Ulrike Sattler, and Frank Wolter. Integrating Description Logics and Action Formalisms: First results. In *Proceedings of AAAI 05*, pages 572–577, 2005.

[5] Franz Baader and Werner Nutt. Basic description logics. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[6] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[7] P. Buitelaar and P. Cimiano. *Ontology learning and population: bridging the gap between text and knowledge*. IOS Press, 2008.

[8] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. Ontology-based database access. In *SEBD*, pages 324–331, 2007.

[9] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *dl-lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[10] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept

languages. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'91)*, pages 151–162. Morgan Kaufmann, Los Altos, 1991.

[11] Corinna Elsenbroich, Oliver Kutz, and Ulrike Sattler. A Case for Abductive Reasoning over Ontologies. In *Proceedings of OWLED 2007*, 2007.

[12] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Chorco. *Ontological Engineering*. Springer, 2004.

[13] T.R. Grüber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[14] Udo Hahn and Stefan Schulz. Ontological foundations for biomedical sciences. *Artificial Intelligence in Medicine*, 39(3):179–182, 2007

[15] Ian Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. PhD thesis, University of Manchester, 1997.

[16] Ian Horrocks. Implementation and optimization techniques. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 306–346. Cambridge University Press, 2003.

[17] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, Bernardo Cuenca-Grau, and James Hendle. Swoop: A Web Ontology Editing Browser. *Journal of Web Semantics*, 4(2), 2005.

[18] Kevin Lee, Thomas Meyer, Jeff Z. Pan, and Richard Booth. Finding Maximally Satisfiable Terminologies for the Description Logic $\mathcal{ALC}$. In *Proceedings of AAAI 06*, pages 269–274, 2006.

[19] Thomas Meyer, Kevin Lee, and Richard Booth. Knowledge integration for description logics. In *Proceedings of AAAI05*, pages 645–650, 2005.

[20] Ralf Möller and Volker Haarslev. Description logic systems. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 282–305. Cambridge University Press, 2003.

[21] Bernhard Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1990.

[22] Bernhard Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43:235–249, 1990.

[23] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. Linking data to ontologies: The description logic DL-LiteA. In *Proc. of the 2nd Workshop on OWL: Experiences and Directions (OWLED 2006)*, 2006.

[24] *The Protégé Ontology Editor*. http://protege.stanford.edu/.

[25] Ulrike Sattler, Diego Calvanese, and Ralf Molitor. Relationship with other formalisms. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 137–177. Cambridge University Press, 2003.

[26] Stefan Schlobach. Diagnosing terminologies. In *Proceedings of AAAI05*, pages 670–675, 2005.

[27] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

[28] Rolf Schwitter, Anne Cregan, and Thomas Meyer. Sydney OWL syntax - towards a controlled natural language syntax for OWL 1.1. In *Proceedings of OWL-ED 2007, OWL Experiences and Directions, Third International Workshop*, 2007.

[29] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2), 2007.

[30] Boontawee Suntisrivaraporn, Franz Baader, Stefan Schulz, and Kent Spackman. Replacing SEP-Triplets in SNOMED CT using Tractable Description Logic Operators. In *Proceedings of AIME 2007*, 2007.

[31] Matt Williams and Anthony Hunter. Harnessing ontologies for argument-based decision-making in breast cancer. In *Proceedings of the International Conference on Tools with Artificial Intelligence (ICTAI 2007)*. IEEE Press, 2007.

[32] Katherine Wolstencroft, Andy Brass, Ian Horrocks, Phil Lord, Ulrike Sattler, Robert Stevens, and Daniele Turi. A little semantic web goes a long way in biology. In *Proceedings of the 2005 International Semantic Web Conference (ISWC 2005)*, LNAI. Springer, 2005.