Reasoning over Individuals: the Instance Store

Lei Li

October 17, 2003

1 Motivation

Description Logics (DLs) are a well-known family of knowledge representation formalisms based on the notion of concepts and roles. They are mainly characterised by constructors for building complex concepts (and roles) from simpler ones. For many DLs, the *subsumption* (sub-class/super-class) relationship among these concepts is defined by their formal semantics and can be decided through adopting a suitable algorithm [7].

The usage of DLs in knowledge representation has been highlighted by the recent explosion of interest in the so-called "Semantic Web" [2]. Claimed to be the next generation World Wide Web, the Semantic Web is trying to enable Web *resources* to be accessible to automated processes using a semantic markup mechanism which can annotate the metadata (data about data) of the Web resources, such that data on the Web will be not only machine readable (just like the Web nowadays is), it will be machine understandable. In the Semantic Web, DLs are set to provide both formal underpinnings and automated reasoning services for Web knowledge representation languages such as OWL [4]—a Web Ontology Language which is now a W3C candidate Recommendation.

With the increasing demand for Semantic Web applications, research in reasoning with individuals is likely to be much more important. For example, after translating annotations from (Semantic) Web pages in OWL into instances of ontology classes (DL concepts), an inferencing service would be useful which is provided in order to retrieve annotations/pages "matching" a given DL concept description or to find out the implicit information among them. Recalling the correspondence between OWL and DLs, it is reasonable then to consider of using ABox reasoning (the DL term for reasoning with individuals) to achieve this.

However, this task seems to be difficult for DLs alone: the difficulty of developing practical ABox reasoning algorithms does not arise only from the complexity of dealing with individuals[1], but from the fact that there is no prior knowledge of how large

[1] Actually, for many DLs, the computational complexity of ABox reasoning remains the same as TBox reasoning.

the size of ABox can be, it might reasonably be growing extremely large. On the one hand, when facing large amounts of individuals, even the state-of-art optimised ABox reasoner RACER [8] will not be able to cope. On the other hand, efficient management of very large volumes of data is well known to be a stronghold of relational databases. The idea of combining relational databases with Description Logics to deal with large amounts of data was, therefore, proposed in [1], namely, the *instance store*.

Intuitively, the main idea of the *instance store* is reducing assertional (ABox) reasoning to terminological (TBox) reasoning and using a database to manage the potential large volume of descriptions of individuals. If an ABox is known to be *role-free*[2], the problem of verifying the satisfiability of that ABox is reducible to the concept satisfiability problem [9], which can then be handled using an existing optimised TBox reasoner such as FaCT [6]. Consequently, standard ABox reasoning tasks like *retrieval*[3] and *realisation*[4] can be answered using a combination of queries against the database and *subsumption*/*classification* requests to the TBox reasoner with TBox reasoning being kept to a minimum. Inspired by vision of this basic *instance store*, it was then decided to follow up this DB+DL approach and try to achieve its extension in several ways.

2 Research Aims

Generally, the problem we are facing is how to bridge the gap between theory and practice: to make an ABox reasoning procedure of practical usage, i.e., terminating in theory is not enough, it must terminate in a reasonable duration while maintaining its soundness and completeness. For example, by applying certain syntactic restrictions to the general ABox formalism, we may obtain a problem which is a special case of the original reasoning task and may have nicer computational properties compared to the original one.

The basic *instance store* can be seen as the first step toward this bridging task, but the restriction is really serious: ABoxes must be *role-free*. The aim of this project is to try to investigate less strict restrictions: improving the existing basic *instance store* by adding some optimisations and relaxing several crucial assumptions made in the basic version. Currently, the basic *instance store* has the following limitations:

1. The implementation is not highly optimised.

2. Only one description can be declared for each individual, and this description is not changeable.

3. Declaring relationships between known individuals are not allowed (i.e., only *role-free* ABoxes are considered.)

[2] An ABox is said to be role-free if it does not contain role assertions, i.e., all the assertions are of the form $a : C$ rather then the form $\langle a, b\rangle : R$

[3] Which individuals are instances of a given concept description?

[4] What are the most specific concept names in the TBox that a given individual is an instance of?

4. The query interface is quite simple, i.e., it provides only two kinds of questions: *Retrieval* and *Realisation*.

5. The ontology (background knowledge) is assumed to be static.

Limitation 2, 3 and 5 are clearly violated in reality and oversimplifying the problem, and relaxing limitation 1 and 4 would make an *instance store* more practical. We believe that most of these limitations can be either removed or relaxed by the following proposed solutions respectively:

1. Some optimisations such as "lexical normalisation" can be applied.

2. In fact, to assert an individual with several descriptions can be treated as a special case of changing the description of an individual (since one can make logical conjunction of all descriptions). The trivial solution is to remove all tuples related to this individual in the *instance store* and assert it with the new description.

3. For various DLs, a technique, so-called *precompletion*, can be used to transform a general ABox (allowing role assertions) into an equivalent *role-free* one, so that relationships between individuals can be allowed. However, when the language supported in the ABox becomes more expressive, e.g., providing concept disjunction and inverse roles, extra restrictions for the standard *precompletion* procedure need to be carefully studied and applied.

4. Previous query languages such as DQL [3] and DL conjunctive queries [10], plus the possible query answering extension when relationships between individuals are to be expressed, can make the query interface more sophisticated and user-friendly.

5. Generally speaking, removing information from the ontology is hard, while adding information can be slightly easier. Techniques like those developed in the field of *truth maintenance* [5] should be helpful.

Proposed solutions including 1, 2, 3 and 4 have already been partially investigated during the first year, and some preliminary results have already shown their promising aspects. Of course, the correctness of solution 3, 4 and 5 remains to be proved formally, and proposed restrictions need to be comprehensible to users.

3 Research Timetable

In accordance with the research aims addressed above, Figure 1 and Table 1 show the proposed research timetable and the research task list respectively.

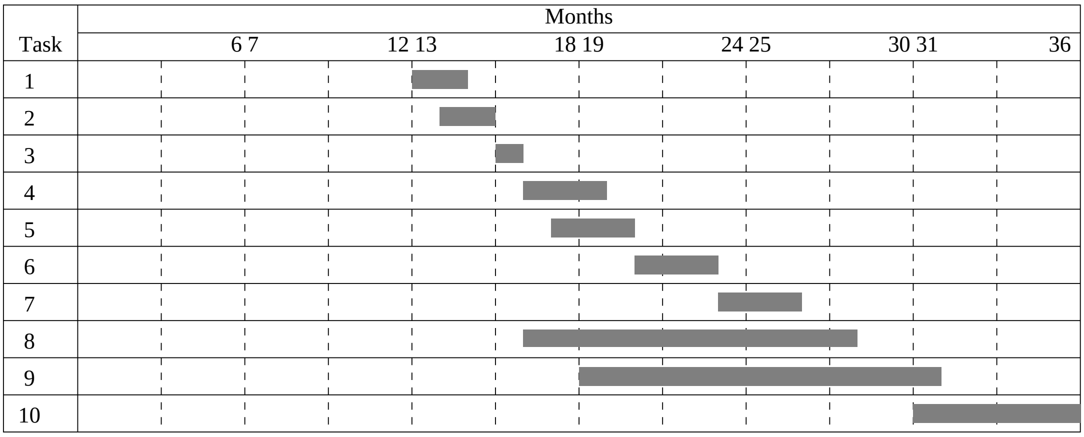

Figure 1: Research Timetable

Task	Duration	Phase Description
1	September-October 2003	Write up a technical report for the basic *instance store*.
2	October-November 2003	Produce analyses and sketched proof for each proposed solutions; a discussion should be made with the supervisor w.r.t. the feasibility of each solution.
3	December 2003	Investigate general optimisations for *instance store*, including lexical normalisation and the unchangeable restriction to descriptions.
4	January-March 2004	Formal proof for *precompletion* procedure of DL $\mathcal{SHIF}(\mathbf{D})$ and clarify what kind of restrictions need to be applied.
5	February-April 2004	Investigate further possible extension for *precompletion* procedure of DL $\mathcal{SHIN}(\mathbf{D})$
6	May-July 2004	Investigate how to enrich the query answering interface. If novel approaches are used, a formal proof should also be presented.
7	August-October 2004	Investigate how to apply truth maintenance technique into *instance store* such that the ontology can be updated; a formal proof should be presented.
8	January-December 2004	Implementation.
9	March 2004-April 2005	Collect test data and carry out evaluation.
10	March-August 2005	Final analysis and thesis write-up.

Table 1: Research Task List

4 Contributions Anticipated

The anticipated main contributions of this research can be summarised in the following points:

- Demonstrate the applicability of the *instance store* in realistic applications (under certain assumptions);
- An evaluation of the techniques used in the *instance store* against other DL/DB or, if one exists, some other DL+DB applications for supporting Semantic Web annotations reasoning;
- A truth maintenance strategy for the *instance store* to cope with the situations when ontologies are changed;
- An extension of the *instance store* technique to relax some of the constraints on the kinds of ABox that can be handled;
- A more sophisticated query answering interfaces for the *instance store* to enable users to define more interesting queries.

5 Anticipated Thesis Structure

The following defines a preliminary structure for the Ph.D. dissertation, obviously it will be refined and improved over time:

- Chapter one: **Introduction** introduces the problem, the area of research to the reader and explains the importance of the problem; the aim and contribution of the thesis is stated and some existing approaches are mentioned.
- Chapter 2: **Background** gives a formal description of the Description Logics; describes some existing DB+DL approach including the basic *instance store* and distinguishes the difference between those and our approach.
- Chapter 3: **Combing Description Logics and Database** presents the rationale behind the new *instance store*, formal proof is given for each relaxed limitation.
- Chapter 4: **Implementing an Instance Store** depicts how to apply the technical results from last chapter to real application, optimisations are addressed in this chapter.
- Chapter 6: **Evaluation** describes the experiments we will perform using a new *instance store*, test data from the developing Semantic Web will be used.
- Chapter 7: **Conclusion and Discussion** draws conclusions from this work and present further directions of research suggested by the results achieved so far.

References

[1] S. Bechhofer and I. Horrocks. A simple DL instance store. 2002.

[2] T. Berners-Lee. The semantic web as a language of logic, 1998.

[3] Daml query language (DQL): Abstract specification, 2002.

[4] M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL web ontology language 1.0 reference, July 2002. `http://www.w3.org/TR/owl-ref/`.

[5] R. E. Filman. Reasoning with worlds and truth maintenance in a knowledge-based programming environment. *Communications of the ACM*, 31(4):382–401, 1988.

[6] I. Horrocks. Using an expressive description logic: FaCT or fiction? In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98)*, pages 636–647, June 1998.

[7] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99)*, number 1705 in Lecture Notes in Artificial Intelligence, pages 161–180. Springer-Verlag, 1999.

[8] http://kogs-www.informatik.uni-hamburg.de/˜race/.

[9] S. Tessaris. Towards an efficient description logics system with individuals. *Technical report, Computer Science Dept. - University of Manchester*, 1998.

[10] S. Tessaris. *Questions and answers: reasoning and querying in Description Logic*. PhD thesis, University of Manchester, 2001.