Description Logic-Ground Knowledge Integration and Management

Joseph B. Kopena*
tjkopena@cs.drexel.edu
Geometric and Intelligent Computing Laboratory
Department of Computer Science, Drexel University
Philadelphia, PA.
http://gicl.cs.drexel.edu/

Summary

This abstract describes ongoing work in developing large-scale knowledge repositories. The project addresses three primary aspects of such systems: integration of knowledge sources; access and retrieval of stored knowledge; scalable, effective repositories. Previous results have shown the effectiveness of description logic-based representations in integrating knowledge sources and the role of non-standard inferences in supporting repository reasoning tasks. Current efforts include developing general-purpose mechanisms for adapting reasoning algorithms for optimized inference under known domain structure and effective use of database technology as a large-scale knowledge base backend.

Knowledge Repositories

Modern scientific and engineering efforts rely on large volumes of data and knowledge. It is therefore important to support such efforts with *knowledge repositories*, significant collections of domain knowledge and data collected from disparate sources and providing a variety of access mechanisms [1]. This goal faces a number of challenges, including:

- *Integration.* Repositories may, and ideally would, collect data and knowledge from many sources. These range from regular, automated input from sensor feeds and databases to sporadic, user-driven contributions from a wide variety of applications. Many will focus on different aspects of the domain, creating a heterogenous mix of knowledge in the repository. In addition, sources that do provide similar input will not be designed *a priori* to function collectively. Input to the repository must be integrated in order to develop a cohesive and useful whole.
- *Access and Management.* Repository users must be supported by a number of access and utilization mechanisms to cope with its volume and complexity. This requires interfaces beyond simple keyword querying services, driven by more sophisticated analysis and knowledge discovery, e.g. browsing through multiple categorizations based on the data rather than labels. Similarly, management tasks, such as maintaining those categorizations, must be provided computationally to be practical.
- *Scalability.* Just as the complexity and volume of repository knowledge make it necessary to provide automated access and management tools, they also challenge the ability to provide sophisticated services. Reasoning mechanisms providing those capabilities must scale to support large quantities of data and a inter-relationships.

Engineering design has been used in this work as the context in which to study these problems. Many engineers spend large portions of their time searching legacy data and catalogs in pursuit of existing solutions which can be modified or assembled to solve new problems. In current practice these resources are often inadequate for working with the large bodies of design knowledge involved. Browsing is typically based on manually-constructed categorizations which are error prone, difficult to maintain, and based on an insufficiently dense hierarchy. Search is limited to keyword scanning or matching on overly simplistic attributes.

This effort has focused on developing techniques and algorithms for *design repositories* (Szykman *et al.* 2000; Szykman, Sriram, & Regli 2001), an evolution of traditional design databases that aim to overcome these limitations through the application of knowledge representation . Function, behavior, rationale, and other aspects of a design are captured and reasoned on to enable search, categorization, analysis aids, and other tasks in support of the engineer.

Representation, Reasoning, and Scalability

The approach taken in this work is rooted on domain ontologies expressed in description logics of limited expressivity, e.g. $\mathcal{ALEN}$. These ontologies are used in integrating knowledge sources; supporting and defining repository inferences; and optimizing reasoning based on domain structure as defined by the ontology. Previously, this work has developed ontologies of engineering function and associated repository reasoning mechanisms (Kopena & Regli 2003b). The representation captures functions and flows in electromechanical designs and is based on informal representations developed in engineering design research (Szykman, Racz, & Sriram 1999; Hirtz *et al.* 2001). Repository contributions are annotated with a model of the artifact under this ontology.

*This work advised by William C. Regli, of the Departments of Computer Science and Mechanical Engineering.

[1] http://nsf.gov/pubs/2004/nsf04528/nsf04528.htm

Such logics are not expressive enough to capture the semantics of science and engineering domains. However, they can describe many structural properties of the domain's objects and define formally valid inferences. This aids in the exchange of design knowledge, enabling input to be annotated with a shared representation used by the repository to integrate contributions. In addition, these semantics define and lend themselves to a number of repository inferences. Basic classification defines a sophisticated search mechanism based on properties of objects and the relationships between them. New applications of non-standard inferences such as most specific class (Küsters & Molitor 2000) and least common subsumer induction (Küsters & Molitor 2001) have also been shown in this work. These inferences provide for and specify a number of repository operations such as query-by-example and knowledge-based categorization.

Currently, this work is addressing the scalability of such reasoning. In addition to their accessible, object-oriented nature, description logics are popular in large part because of their decidability and efficient reasoning in practice. However, description logic reasoners are still challenged when knowledge bases approach the size envisioned of realistic, useful knowledge repositories. This concern is being approached in this work along three complementary directions:

- *Parameterized reasoning.* This project has also developed OWLJessKB (Kopena & Regli 2003a), a description logic reasoner for the Ontology Web Language (OWL) [2]. OWLJessKB implements sound but incomplete description logic reasoning within an expressive production system. Initially, gross decisions were made as to which inferences to support. Presumed unlikely and inefficient inferences, such as classification of transitive properties from closed role extensions, were excluded.

  This notion is currently being explored in more depth in the form of experiments evaluating fine-grained, run-time control over supported inferences, allowing applications to balance between expressiveness and efficiency. Analytical results for such tradeoffs are the foundation of description logic research; this effort focusses on empirical runtime benefits at a lower level of control. Examples include supporting only the necessary direction of a class expression or cardinality restrictions of solely 0 and 1.

- *Domain-adapted algorithms.* Similarly, this work is also exploring automated compilation of reasoning algorithms adapted to given domain structure. This effort is particularly focused at specializing algorithms for non-standard description logic inferences. The goal is to optimize generic specifications of the algorithms based on the known domain ontology and statements made in the input. For example, some checks and components in most specific class and least common subsumer induction may be eliminated if the ontology and input adhere to certain structure and limitations, e.g. acyclic instance data or a finite, *a priori* vocabulary of classes.

- *Database backends.* It is likely that either of the above approaches alone will be quickly overcome by the volume of elements in a repository. In particular, it will consume large amounts of memory and involve quite large data structures. To address this, these approaches are being combined and adapted to use database technology for the underlying storage. This provides a persistent, out-of-core, scalable knowledge base for repository reasoning. In contrast to current work in this area, e.g. (Horrocks *et al.* 2004), the repository requires a more powerful input language and must support a wider variety of inferences.

[2] `http://www.w3.org/TR/owl-ref/`

Conclusion

It is hoped in studying these approaches in the context of design repositories that general methods and tools may be developed to meet the challenges of large-scale knowledge integration, reasoning, and management. This is a necessary task in constructing knowledge repositories infrastructure to support increasingly complex, distributed, and voluminous scientific and engineering efforts in a variety of domains.

More information and previous publications are available at `http://edge.cs.drexel.edu/assemblies/`.

References

Hirtz, J.; Stone, R.; McAdams, D.; S, S.; and Wood, K. 2001. Evolving a functional basis for engineering design. In *ASME DETC/13th Design Theory and Methodology*. New York, NY, USA: ASME. DETC01/DTM-21688.

Horrocks, I.; Li, L.; Turi, D.; and Bechhofer, S. 2004. The instance store: DL reasoning with large numbers of individuals. In *The Description Logic Workshop*, 31–40.

Kopena, J. B., and Regli, W. C. 2003a. DAMLJessKB: A tool for reasoning with the semantic web. In *Proc. of the 2003 Int. Semantic Web Conf.*, 628–643. Springer Verlag.

Kopena, J. B., and Regli, W. C. 2003b. Functional modeling of engineering designs for the semantic web. *IEEE Data Engineering Bulletin* 26(4):55–62.

Küsters, R., and Molitor, R. 2000. Computing most specific concepts in description logics with existential restrictions. Technical Report LTCS 00-05, Research Group for Theoretical Comp. Science, Aachen Univ. of Technology.

Küsters, R., and Molitor, R. 2001. Computing least common subsumers in $\mathcal{ALEN}$. In Nebel, B., ed., *17th Int. Joint Conf. on AI*, 219–224. Morgan Kaufmann.

Szykman, S.; Bochenek, C.; Racz, J. W.; Senfaute, J.; and Sriram, R. D. 2000. Design repositories: Engineering design's new knowledge base. *IEEE Intelligent Systems* 15(3):48–55.

Szykman, S.; Racz, J. W.; and Sriram, R. D. 1999. The representation of function in computer-based design. In *ASME DETC/11th Conference on Design Theory and Methodology*. New York, NY, USA: ASME. DETC99/DTM-8742.

Szykman, S.; Sriram, R. D.; and Regli, W. C. 2001. The role of knowledge in next-generation product development sy stems. *ASME Transactions, the Journal of Computer and Information Sc ience in Engineering* 1(1):3–11.