ONTOSERVER — Infrastructure for the Semantic Web

Raphael Volz
Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
volz@aifb.uni-karlsruhe.de
http://www.aifb.uni-karlsruhe.de/WBS

1 Motivation

Ontologies build the conceptual backbone of the Semantic Web. By providing a shared and common understanding of a domain they can be communicated across people and application systems for facilitating knowledge sharing and exchange. In the context of the Semantic Web ontologies describe domain theories for the explicit representation of the semantics of relational (meta-)data.

As ontologies aim at consensual domain knowledge their development requires a cooperative process which has to be supported by central repositories that enable persistent storage and concurrent access. We believe that many Semantic Web applications are also charaterized by concurrent access (i.e. building a Web portal to support the Semantic Web Research Community[1]). Part of the Semantic Web infrastructure currently implemented at our institute is ONTOSERVER, a multi-user capable metadata and ontology repository that uses external inference engines to provide model checking and querying.

In our design we identified the following general requirements for a central repository:

- **Persistence** A repository must prevent data loss by providing persistent storage of data.
- **Update semantics** Data must be updateable, users must be able to provide new data (concering ontology schema data as well as ontology instance data)
- **Concurrency** A repository must be capable of providing concurrent access to its data.
- **Security** In a networking environment the feature of security is of very high importance.

In the Semantic Web the technology of choice for exchanging and storing ontologies is the Resource Description Framework (RDF). Its abstract data model provides a fundamental standard for the Semantic Web and especially appeals because of its generality and extensibility.

RDF can be used to represent all data occurring in ontological applications (cf. figure 1):

1. **Representation Vocabularies** that describe ontology languages (like DAML+OIL[2], DRDFS [DFZD01] or the RDF Schema proposal [BG99]) and usually have semantics based on a logical theory that can be interpreted by a inference engine.
2. **Ontologies** that use a specific representation vocabulary and represent a domain theory.
3. **Relational Metadata** that uses the vocabulary specified by an ontology and provides instance data for the domain model.

[1] You are welcome to visit http://www.ontoweb.org/

[2] cf. http://www.daml.org/2001/03/daml+oil-index

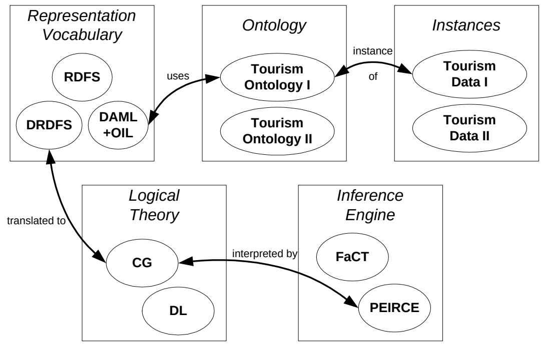

Figure 1: Representation Vocabularies, Ontologies and Instances

Recently, RDF repositories like RDFdb[3], RDFSuite[4] or Sesame[5] among others have been developed. These systems can be used as a starting point to implement the above mentioned requirements. In fact, due to RDF's generality RDF frameworks in general can be reused across a wide range of Semantic Web applications to provide generic data access and storage components.

2 Why a RDF repository is not enough!

Although it is possible to store all data (representation vocabularies, ontologies and relational (meta-)data) related to a Semantic Web application in one "bag", the distinction of this data is of very high importance in applications, especially in ontology engineering scenarios. A typical task in these scenarios is performing consistency checks of ontologies. This involves structural and semantic checking of the conformance

[3] cf. http://www.guha.com/rdfdb/

[4] cf. http://www.ics.forth.gr/proj/isst/RDF/

[5] cf. http://sesame.aidministrator.nl/

of a given set of relational (meta-)data towards an ontology as well as conformance of an ontology towards a given representation vocabulary. This tasks poses special challenges to RDF repositories as a repository would have to understand the formal semantics of a given representation vocabulary to perform this task. Other tasks that require understanding of semantics are inferencing and querying of RDF data. Interestingly these common tasks are not available in current RDF repositories.

When provided these services should also be scalable and performant. Usually this implies usage of highly optimized (thus specific) structures as found in today's inference engines. This is also true for database technology, where a second important property can be identified: Usually any ontology modification (comparable with schema modification in databases) can only happen when the part of the database affected by the change is not available to users. This is due to the fact that ontology (= schema) modification normally implies data modification. Also, optimization techniques usually depend on schema and access structure (for instance choosing hash tables or b-trees for indexing).

As the number of possible representation vocabularies whose semantics would have to be understood by a repository is infinite we conclude that it is not possible for a single repository to provide this for all possible representation vocabularies.

Therefore, we propose to build a comprehensive infrastructure around a basic repository. We believe that it is beneficial (especially for less tractable representation vocabularies) to use external inference engines (like SilRI [DBSA98] or FaCT [Hor99]) to deploy a given RDF model whose ontology (schema) is fixed[6]. Then, a specific inference engine could be used for querying and checking of given RDF models. In general, the integration of such external inference engines should be hidden from the users of such infrastructure, as this setup clearly is an administrator task. To implement this, we are dynamically registering inference engines to the XML namespace of given representation vocabularies and allow to dynamically locate service descriptions that wrap engine specific interfaces to ONTOSERVER access interfaces.

3 ONTOSERVER – Infrastructure for the Semantic Web

Due to lack of space, we can only briefly mention the rationale that lead our architecture of ONTOSERVER. The general design reflects the reusability of ONTOSERVER's components for stand-alone Semantic Web applications. Therefore we introduced three levels:

1. The application framework level contains all components that can be reused in non-server applications, specifically data access and storage components as well as external service connectors.
2. The server framework level contains additional components specific to server applications, this includes user/rights/security management and transaction facilites.
3. The Application level contains custom applications and provides one prototypical implementation: ONTOSERVER itself.

[6] The reader may note, that this could also be the quiescent state (version) of a submodel (view) that has been tailored to meet the requirements of a certain application or audience.

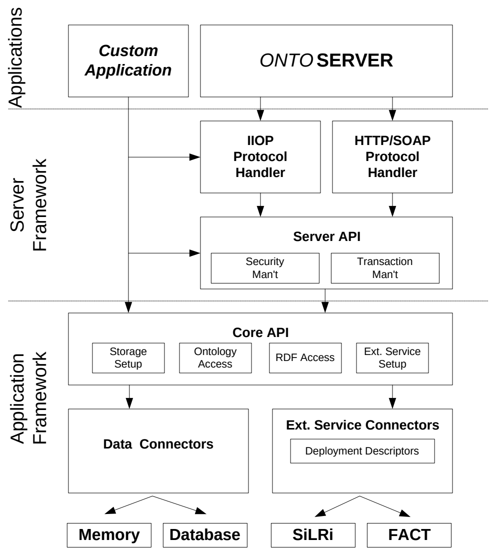

Figure 2: ONTOSERVER: General architecture

4 Related Work

Protégé is an ontology engineering system that supports the flexible definition and instantiation of different representation vocabularies has been introduced in [NSD+01]. However the internal data model does not build on RDF, thus it's components can not be reused to build Semantic Web applications. Interestingly, the formal semantics of representation primitives can only be provided outside the tool, to do this users must write plug-ins. In the future Protégé's plug-in mechanism opens the possibility to use Protégé as an editing client for ONTOSERVER.

5 Future Work

A general challenge is the interoperability between different representation vocabularies and concrete ontologies along with translation of relational metadata instances. As [DvHB+00] mentioned, RDF's "Subject-Predicate Structure leads to independance of objects". It is therefore technically possible to provide mappings between two RDF models, a feature would that would ultimately leverage the Semantic Web.

Another challenge is posed by the fact that ontologies are rarely stable following initial creation especially in dynamic settings like the Web, where content changes with light-speed. Therefore versioning and controlled evolution of metadata is a requirement for the Semantic Web.

Last but not least many of our applications require to have views on metadata that present virtual subsets of one model to special audiences.

References

[BG99] D. Brickley and R.V. Guha. Resource description framework (RDF) schema speci£cation. Technical report, W3C, 1999. W3C Proposed Recommendation. http://www.w3.org/TR/PR-rdf-schema/.

[DBSA98] S. Decker, D. Brickley, J. Saarela, and J. Angele. A Query and Inference Service for RDF. In *Proceedings of the W3C Query Language Workshop (QL-98), http://www.w3.org/TandS/QL/QL98/, Boston, MA, December 3-4*, 1998.

[DFZD01] Alexandre Delteil, Catherine Faron-Zucker, and Rose Dieng. Extension of rdfs based on the cgs formalism. In *Conceptual Structures - Broadening the Base. Proc. ICCS 2001. Lecture Notes in Arti£cial Intelligence 2120.*, pages 272–285, 2001.

[DvHB+00] S. Decker, F. van Harmelen, J. Broekstra, M. Erdmann, D. Fensel, I. Horrocks, M. Klein, and S. Melnik. The semantic web: The roles of XML and RDF. *IEEE Internet Computing*, 4(5), 2000.

[Hor99] I. Horrocks. FaCT and iFaCT. In *International Workshop on Description Logics (DL'99)*, pages 133–135, 1999.

[NSD+01] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protege-2000. *IEEE Intelligent Systems*, 16(2), 2001.