LDAP-based Ontology for Information Integration

Lule Ahmedi*, Pedro José Marrón and Georg Lausen

Universität Freiburg
Institut für Informatik
Georges-Köhler-Allee, Geb. 51
79110 Freiburg, Germany
{ahmedi,pjmarron,lausen}@informatik.uni-freiburg.de

Abstract. The increase in popularity of the Internet has lead to the proliferation of heterogeneous information systems that could be better utilized if they operated under a common query interface. The purpose of this paper is to describe the representation formalism of ontologies in an LDAP-based information integration system, and to point out the advantages of such an approach over traditional systems. The simplicity, coherence and uniformity of the LDAP model allows us to seamlessly integrate source data, schemata discrepancies, and semantic information under a common framework, that is, by design, able to reconcile integration and data processing issues.

1 Introduction

Since the conception of the LDAP protocol version 3 in 1997 [WHK97], the use of lightweight directories to store information has been steadily gaining momentum. Today, many universities and research centers, like AT&T, use LDAP servers as a means to manage information about their members, organizations, networks, etc., and companies like Netscape or Microsoft offer LDAP support even in their Internet browsers.

At the same time, due to the ubiquity of the Internet, information integration and, particularly, the use of ontologies to ease the task of data integration is receiving more and more attention by the research community [MKSI00,DEFS99,AK93]. With the increase in popularity of LDAP servers, traditional problems in the field of information integration should be revisited to accommodate the new technology, especially, if this technology is so tied to network and distribution channels, as it is the case with LDAP technology that, by design, offers distribution capabilities not present in more traditional databases [WHK97].

In this paper, we present a new integration approach, targeted to the ontologies model, but adapted to LDAP technology, which promises to be able to

* The work of this author is supported by the Deutsche Forschungsgemeinschaft, Aktenzeichen La 598/4-1.

easily solve all classical integration problems along with the LDAP advantageous network behavior features. In Sect. 2, we give a brief overview of the capabilities offered by LDAP servers, suitable for use in our model. Section 3 goes into the details of our approach, leaving the comparison to other systems for Sect. 4. Section 5 discusses future work, and finally, Sect. 6 concludes this paper.

2 LDAP Overview

An LDAP server can be considered a semistructured database with limited transaction capabilities, made up of the following main components:

Directory Schema: Defines a finite set of classes, attributes and types. Each attribute must have a type, and each class specifies a set of `required` and `allowed` attributes.

Directory Instance: Contains a finite set of entries organized in a forest, where: (1) Each entry belongs to at least one class; (2) has a non-empty set of (possibly) multi-valued attribute-value pairs that conform to the schema definition; (3) defines at least attributes *oc* and *dn*, where *oc* determines what classes the entry belongs to, and *dn* provides a unique distinguished name for the entry; and (4) is placed in the instance hierarchy based on their distinguished name.

In addition, LDAP offers a query facility based on filter definitions consisting of the following four components:

Base: The distinguished name of the entry in the directory instance where the search will start.

Scope: Can be `base`, if the search is to be restricted to just the first node, `onelevel`, if only the first level of nodes is to be searched, or `subtree`, if all nodes under the base should be considered by the filter expression.

Filter Expression: Defined as the boolean combination of atomic filters of the form $(a\ op\ t)$, where: a is an attribute name, op is a comparison operator out of the set $\{=, \neq, <, \leq, >, \geq\}$, and t is an attribute value.

Projection: Defines the set of attributes to be returned.

For more detailed information about the LDAP protocol, see either [HSG99] for an informal description, or [JLM$^+$99] for a formal one. For the purpose of this paper, we are mostly interested in the schema definition capabilities, which we will use to model ontologies, as discussed in the next section.

3 Ontology Model

3.1 Basics

According to Gruber [Gru93], an **ontology** is an explicit specification of an abstract, simplified view of the world of interest, also called a conceptualization. Information integration systems often make use of ontologies to specify a global conceptualization that describes both, a set of terms and the relationships between them.

The integration system provides, via the ontology definition, a common query domain for all users that abstracts the underlying heterogeneous sources, independently of the particular query model each sub-system defines. All queries performed at the domain level are answered by reformulating them into source-level queries, and then transforming the results into the terms of the unified view.

The LDAP model described in the previous section has support, not only for the integration process we just referred to, but also for the standard ontology design requirements, namely:

Modeling capabilities: LDAP provides primitives for concepts (by means of classes), roles (by means of attributes), key roles, subclass relationships and general constraints.

Flexibility and Maintainability: Changes to an LDAP-based ontology are easily made, allowing for the upgrade of concepts and roles in a flexible manner, as opposed to fixed ontology systems.

Manipulation facilities: LDAP provides manipulation operators on top of which operators aimed at integration can be defined.

Schema querying: Since the LDAP data model is self-describing, concept and role names are allowed as part of the result of a query.

Proper design techniques and important issues for the creation of ontologies for heterogeneous sources are discussed elsewhere [Usc96,UG96], and are out of the scope of our work. The purpose of this paper is to show in detail, and as a particular aspect of the breadth of functionality of our LDAP-based integration system, how ontologies can be represented in the LDAP model.

3.2 LDAP Representation Formalism

As a running example to describe in more detail our representation, we use the Mondial ontology extracted from the Geography subontology in CYC [LG90]. Figure 1 provides a textual representation that uses indentation to indicate the subconcept relationship and parenthesis to indicate properties; Fig. 2 provides its (partial) LDAP-based representation constructed following the formalisms described in this section.

The main constructs of an ontology, namely *concepts* and *roles*, together known as *terms*, are introduced in our LDAP-based ontology as subclasses of the special `top` class with the following structure:

```
term OBJECT-CLASS ::= {
   SUBCLASS OF {top}
   MUST CONTAIN {oc,name}            // required attributes
   TYPE oc OBJECT-CLASS
   TYPE name STRING }
concept OBJECT-CLASS ::= {
   SUBCLASS OF {term}
```

```
Geography(Continent,GeopoliticalEntity,Mountain,BodyOfWater,Desert,Island)
Continent(name,area)
GeopoliticalEntity(name,population)
  Country(area,car_code,population_growth,infant_mortality,gdp_total,
          gdp_agri,gdp_ind,gdp_serv,inflation,government,encompassed,
          ethnicgroup,religion,language,border,StateGeopolitical,City)
    IndependentCountry(indep_date)
  StateGeopolitical(area,City)
  City(longitude,latitude,located)
```

Fig. 1. Mondial ontology

```
       MAY CONTAIN {subclass_of,superclass_of,filter,link,
                    synonym,hypernym_of,hyponym_of}
                                            // allowed attributes
       TYPE subclass_of,superclass_of DN(concept)
       TYPE filter LDAP-Filter
       TYPE link DN(source term)
       TYPE synonym,hypernym_of,hyponym_of DN(concept) }
    role OBJECT-CLASS ::= {
       SUBCLASS OF {term}
       MUST CONTAIN {domain,range}
       TYPE domain DN(concept)
       MAY CONTAIN {key,link,synonym,hypernym_of,hyponym_of}
       TYPE range DN(concept), STRING
       TYPE key {YES}
       TYPE link DN(source term)
       TYPE synonym,hypernym_of,hyponym_of DN(role) }
```

Each class contains a set of attributes, which can be categorized as either *basic schema constructs*, *concept hierarchy constructs*, *integration constructs* or *interontology relationship constructs*. The purpose of these attributes is explained in the sequel.

Basic Schema Constructs. The basic schema constructs define properties that are fundamental for every LDAP-based ontology schema, namely: the `oc` and `name` attributes, that are required and represent the class (or classes) the entry belongs to, and the name of the entry, respectively; the `domain` and `range` attributes, that, for a *role* declaration, specify the concept it belongs to and the type of its values respectively; the `key` attribute, that indicates whether or not a given *role* can uniquely identify a concept for which it is defined; and the `filter` attribute that specifies general constraints imposed on a particular *role* or *concept* to define the so-called **defined terms**.

For example, in Fig. 2, the `car_code` node is defined by stating that it belongs to the class `oc=role`, its `domain` is `Country`, its `range`, any string, and it is a `key` of `Country`. As an example of a **defined term** definition, the `NewCountry` node has a constraint of the form `filter=(indep_date > 1990)` that indicates

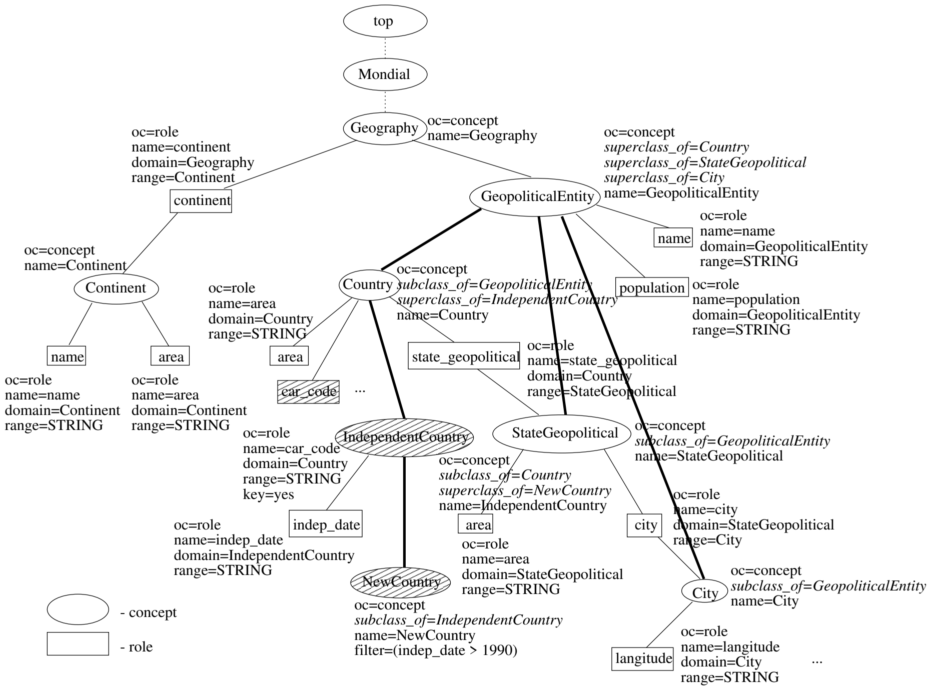

Fig. 2. Mondial ontology in LDAP; shaded components are referred in the text

that only independent countries whose independence date is greater than 1990 are considered `NewCountry`.

Concept Hierarchy Constructs. The `subclass_of` and `superclass_of` attributes express the subclass and superclass relationships between concepts in the system. In Fig. 2, this relationship is additionally depicted by the thicker lines that connect two or more concepts, where, for example, `NewCountry` is defined as a subclass of `Country`.

Integration Constructs. The `link` attribute is used to enable the integration of different information source schemas into a single LDAP directory instance. It indicates that a given `term` in the ontology is terminologically equivalent with the term in the LDAP source model it (its value) references to. Before the annotation of the domain ontology with `link` attributes takes part, the transformation of each source schema into the corresponding LDAP representation should be done. The specific details of the principal architecture of integration and its key part of specifying and considering link attributes, as well as its concrete advantages are out of the scope of this paper, but it is important to emphasize that by using the `link` attribute our system is able to resolve most of the conflicts existing between distinct heterogeneous sources.

Interontology Relationship Constructs. In order to capture the semantic subtleties, differences and similarities among ontologies, we use the `synonym`, `hypernym_of` and `hyponym_of` attributes. The `synonym` attribute indicates that two concepts with different names are equivalent, and should be treated as such, whereas the `hypernym_of` and `hyponym_of` specify a subsumption relationship that defines whether a term is more general than another, as defined by the former, or more specific, as defined by the latter.

Note that the specification of all these constructs including `link` correspondences between ontology and data sources is part of the application design.

4 Feature Comparison

Although extensive work has been done in the area of modeling ontologies for data integration, most researchers use logic-based languages like description logics, F-Logic or ODL to describe them and benefit from the reasoning abilities of well-established logic systems. Our approach, on the other hand, uses a relatively simple model with a clear syntax that allows us to provide a unified formal framework to be used for both, ontology definition and information integration using the standard and well-established LDAP technology. An subsumption reasoning mechanism is not excluded and may be added to the system.

Furthermore, due to the hierarchical and flexible structure of LDAP, we can very easily integrate not only structured data, but also semistructured data, as opposed to systems like Sims [AK93] or Observer [MKSI00] that are limited to the expressive power of their ontology description languages and are only able to deal with relational and flat file databases. Not even systems like Florid [HKL+98], Ontobroker [DEFS99] or Momis [BCV99], which combine the reasoning capabilities of logic systems with the expressive power of an object-oriented model are able to describe XML data as naturally as we do in our system. The lack of a tree-like modeling structure forces them to map XML into an artificial structure not particularly well suited for such graphs, whereas our model is based on a tree structure that resembles that of XML.

Tsimmis [GMPQ+97] addresses the above deficiencies by taking a similar approach to ours, namely using the OEM data model to describe mediated views and sources, as opposed to our LDAP-based information model, which offers, by design, additional benefits derived by the nature of the LDAP directory services. Directory-based stores of data support a ubiquitous Internet access standard, giving a guarantee to almost any user connected to the network to get access to this data.

Some other approaches, like [BGL+99,CCS00] use XML as their common model for data exchange and integration, but fail to consider the use of ontologies as an integral part of the system. Our LDAP-based model, on the other hand, provides seamless integration with ontologies, which allows us to scale our system both from the source and from the user perspective, of paramount importance for the Web.

These advanced features, combined with the fact that the hierarchical LDAP namespace allows us to implicitly distinguish nodes belonging to different source trees without incurring in any additional work, makes LDAP an ideal candidate for the field of integration.

5 Future Work

A preliminary version of the system is already in the works and providing very promising results, as detailed in [ML01]. Nevertheless, the system is far from finished. We plan to continue development on it by making the following additions:

Query Rewriting Operators: To transform the ontology level query to source level queries.

Schematic Reconciliation: To resolve the schematic differences that exist between semantically similar terms in the ontology and in the sources. This involves the definition and implementation of more `link`-like attributes.

Ontology Partitioning: To allow for distribution and replication of ontologies where needed.

Hierarchical Ontology Management: To allow for more complex relationships to be defined among distributed ontologies that go beyond the synonym, hyponym, and hypernym relationships.

6 Conclusion

In this paper, we have presented an LDAP-based representation formalism for ontologies which provides obvious advantages with respect to more classical integration system because of the simplicity, coherence and uniformity of the LDAP formalism.

Furthermore, our system is, to the best of our knowledge, the only system that combines the advantages of a hierarchical data model, on which, for example, XML documents could be mapped, and those of ontologies as the integration basis.

The integration of several relational databases by means of LDAP directories is supported by [Inc00]; however, the problem of integrating heterogeneous objects is not considered, in particular.

Acknowledgments

We would like to thank reviewers for their helpful suggestions.

References

[AK93] Y. Arens and C. Knoblock. SIMS: retrieving and integrating information from multiple sources. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 22(2):562–563, June 1993.

[BCV99] S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. SIGMOD Record, Special Issue on Semantic Interoperability in Global Information, Vol. 28, No. 1, March 1999.

[BGL+99] C. Baru, A. Gupta, B. Ludaescher, R. Marciano, Y. Papakonstantinou, and P. Velikhov. XML-based information mediation with MIX. In Demo Session, ACM-SIGMOD'99, Philadelphia, PA, 1999.

[CCS00] V. Christophides, S. Cluet, and J. Sim'eon. On Wrapping Query Languages and Efficient XML Integration. Proceedings of ACM SIGMOD Conference on Management of Data, Dallas, Texas, May 2000.

[DEFS99] S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In R. Meersman et al., editor, DS-8: Semantic Issues in Multimedia Systems. Kluwer Academic Publisher, 1999.

[GMPQ+97] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117–132, 1997.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Aquisition*, 5(2):199–220, 1993.

[HKL+98] Rainer Himmeröder, Paul-Th. Kandzia, Bertram Ludäscher, Wolfgang May, and Georg Lausen. Search, analysis, and integration of web documents: A case study with florid. In *Proc. Intl. Workshop on Deductive Databases and Logic Programming (DDLP'98)*, pages 47–57, Manchester, UK, 1998.

[HSG99] T. A. Howes, M. C. Smith, and G. S. Good. *Undestanding and Deploying LDAP Directory Services.* Macmillan Network Architecture and Development. Macmillan Technical Publishing U.S.A., 1999.

[Inc00] Radiant Logic Inc. RadiantOne version 1.0. http://www.intelligententerprise.com/000301/products.shtml, 2000.

[JLM+99] H. V. Jagadish, Laks V. S. Lakshmanan, Tova Milo, Divesh Srivastava, and Dimitra Vista. Querying network directories. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadephia, Pennsylvania, USA*, pages 133–144. ACM Press, 1999.

[LG90] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project.* Addison-Wesley, Reading, Massachusetts, 1990.

[MKSI00] E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.

[ML01] Pedro José Marrón and Georg Lausen. HLCaches: An ldap-based distributed cache system for xml. Submitted, 2001.

[UG96] M. Uschold and M. Gruninger. Ontologies: principles, methods, and applications. Knowledge Engineering Review, 11(2), 93–155, 1996.

[Usc96] Mike Uschold. Building ontologies: Towards a unified methodology. In *16th Annual Conf. of the British Computer Society Specialist Group on Expert Systems*, Cambridge, UK, 1996.

[WHK97] M. Wahl, T. Howes, and S. Kille. Lightweight directory access protocol (v3). RFC 2251, December 1997.