Information Integration and the Semantic Web

Laks V. S. Lakshmanan*
University of British Columbia
laks@cs.ubc.ca

Fereidoon Sadri†
University of NC, Greensboro
sadri@uncg.edu

1 Introduction

Information integration and interoperability among information sources are related problems that have received significant attention since early days of computer information processing. Initially, for a few decades, the focus was on integration/interoperability for a relatively small number of sources. This is the setting encountered in traditional business and service applications, for example when two companies merge or several services interoperate (which requires the integration of their information systems). Much of the work in this context of federated or multi-databases focused on integrating schemas by defining a global schema in an expressive data model and defining mappings from local schemas to the global one [19]. More recently, in the context of integration of data sources on the internet, the so-called *global-as-view* (GAV) and *local-as-view* (LAV) paradigms have emerged out of projects such as TSIMMIS [20] and Information Manifold (IM) [12].

Recently, the advent of XML as a standard for online data interchange holds much promise toward promoting interoperability and data integration. The focus has also shifted to one of providing integration and interoperability among a large number of independent and autonomous information sources. But XML, being a syntactic model, in itself cannot make interoperability happen automatically. Two main challenges to overcome are: (i) data sources may model XML data in heterogeneous ways, e.g., using different nestings or groupings or interchanging elements and attributes, and (ii) sources may employ different terminology, a classic problem even in multi-database interoperability. While earlier approaches to integration can be extended to handle XML, they suffer from the significant overhead of having to design a commonly agreed upon global schema. Can we interoperate without this overhead?

Indeed, there are a few recent proposals that do overcome the need for a global schema – Halevy et al. [9] and Miller et al. [15]. In a nutshell, both of these approaches rely on source to source mappings. One problem here is that requiring such mappings for all pairs is too tedious and cumbersome since the mappings, even if generated semi-automatically, still require significant manual intervention to create. In order to mitigate this, one can merely insist, as [9] does, that the graph of pairs with available mappings be connected. A second problem is that when a source s_i is mapped to source s_j, if s_j does not have some of the concepts present in s_i, then they will be lost. E.g., s_i may include the `ISBN` for all its books while s_j may not.

Independently of all this, recently there has been much excitement around the *Semantic Web* [18] initiative, coming as it does with its own host of technologies such as resource description framework (RDF) [16] and ontology description languages such as DAML+OIL and OWL [5, 14]. The promise of the Semantic Web is to expose the semantics of the information content of web resources (including text, audio, video, etc.) thus taking the web to a higher semantic level, enabling easy exchange of data and applications.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

*Supported by NSERC.
†Supported by NSF grant IIS-0083312.

Our work on interoperability and information integration has been motivated by asking how we can take advantage of the Semantic Web initiative. Our thesis is that each source('s administrator) interested in participating in data and application sharing should "enrich" itself with adequate *semantic declarations*, providing a semantic view of the information contents. These declarations would essentially expose the semantic concepts present in the source using common, but possibly application or even community specific terminology, in the spirit of the Semantic Web initiative and frameworks such as RDF. Further infrastructure such as application wide ontologies could exist and help relate terminologies used across different communities within an application domain or even beyond. Queries across data sources so enriched can be composed over the vocabulary consisting of community wide RDF vocabulary or application wide ontology. They can be agnostic to the specific modeling constructs or terminologies employed in the local sources (i.e., below their semantic views). Query processing and optimization is the responsibility of the interoperability system, and is carried out in consultation with ontology servers that provide the common terminologies [11]. We should emphasize that we are not assuming a single standard ontology, not even for a given application. Rather, we expect that, within any one application domain, a limited number of ontologies will become widely used by the interested communities.

Below we provide a summary of of our ongoing research on the X-DARES-U (XML DAta waREhouse with Semantic enrichment at UBC/UNCG) project [10]. We also discuss inherent challenges – both technical and social in making interoperability and information integration possible on the internet scale. The need for a killer application has been realized for some time in the Semantic Web community. We argue that interoperability and integration offer such an application.

2 X-DARES-U Overview

The X-DARES-U approach is based on the existence of an infrastructure constituted by the following components: A semantic view provided by data sources, data mappings to represent source data in its semantic view, existence of ontologies and inter-ontology mappings, tools for inferring integrity constraints over mappings, and query optimization techniques. In this section, we briefly overview each of these.

2.1 Semantic Declarations

Local information sources can be XML documents, relational databases, spreadsheet documents, or servers of other data formats. A local source can join an interoperability effort as long as it can provide semantic declarations and mappings as discussed below. *We should emphasize that these declarations are local.* No overhead in providing a global schema is needed in our approach. This is in contrast to traditional approaches to data integration where significant overhead is incurred in developing a global schema for the federation.

The semantic content of an information source is declared in the form of *concepts* and *properties* represented in the source. This is in the spirit of knowledge representation languages and ontologies that use concepts (or classes) and properties (or relationships) for semantic representation. Each source chooses an appropriate semantic model based on terminologies that are either chosen from a common ontology, or it can create and publish its own terminology. A mapping of the source's native data format to the semantic model is then specified. If necessary, the source also specifies a mapping from its semantic model to a better known ontology. These declarations can be provided in a number of ways, including Semantic Web tools such as RDF Schema [17]. We will use the term *semantic content model* (*semantic model*, for short) of a local source to refer to the semantic declarations, in the form of concept/property constructs, as discussed above.

Example 1 (Semantic Model): Consider a federation of catalog sales stores maintaining their merchandise information in various systems (XML, relational, etc...) and formats. The DTD of one of these stores is shown below (for additional DTDs see the complete example in [11]).

```
<!ELEMENT store (warehouse*)>
<!ELEMENT warehouse (city, state, item*)>
<!ELEMENT item (id, name, description)>
<!ATTLIST warehouse id ID #REQUIRED>
```

A possible semantic model for these sources consists of the concepts `item`, `warehouse`, `city`, `state`, and properties `item-itemId`, `item-name`, `item-description`, `item-warehouse`, `warehouse-warehouseId`, `warehouse-city`, `warehouse-state`. Motivated by RDF, the first parameter of each property is a URI. The second parameter can be a URI or a string. Note that we need to provide the mapping between URIs and database-style identifiers, such as `itemId`. The properties `item-itemId` and `warehouse-warehouseId` do precisely this.

Once the concepts and properties for the declaration of the semantic content of a source are determined, the local DBA or user should provide the mappings from the local data into the selected semantic content model. This, in effect, is providing a *semantic view* of the information content of the source in its semantic model. We envision semi-automated tools that can assist DBA or user in this task [11]. The mappings can be represented succinctly as rules using a combination of XPath and Datalog as in [11]. Providing the mappings in a more practical representation, such as an XSLT code that generates an RDF document (the view) as output, is also possible. The mapping specification tool can be configured to produce a number of different presentations.

Example 2 (Mappings): The mapping rule for the property `item-warehouse` is shown below. The functions f_I and f_W are *URI-generating* functions that produce URIs from database-style IDs. Please refer to [11] for details and additional mappings.

`item-warehouse(`f_I`($I),`f_W`($W))` $\leftarrow$ `source1/store/warehouse $X, $X/item/id $I, $X/@id $W`

2.2 Query Formulation

Consider a federation of information sources enhanced with semantic declarations and mappings as discussed in the previous section. A *global* query is formulated in terms of concepts and properties. The interoperability system is responsible for the optimization and execution of the query. It may require services from ontology brokers and directory servers to reconcile ontological and taxonomic differences in local semantic models as well as local data. For brevity, we suppress this detail and focus on the case of single ontology in the sequel.

As mentioned above, a query is formulated in terms of concepts and properties, which may be at any level of an ontology. Let p be a property mentioned in the query (such as `item-warehouse` in our example). A source i may have this property in its semantic model. If so, it contains a *fragment* of p, which we will denote by p_i. The *global content* of p is the collection of all of its local fragments (p_i). What exactly is the meaning of a *collection* of fragments? We assume, in general, that local sources have *overlapping* data. Hence, *collection* should provide a mechanism to handle the overlap. If data among sources are *consistent*, then the collection is simply a (possibly duplicate eliminating) union. More complicated mechanisms are needed when sources contain inconsistent and/or uncertain data. Here we assume consistency within each source as well as among sources, and use (possibly duplicate eliminating) union as the semantics of collection.

2.3 Architecture, Query Processing and Optimization

A coordinator-based architecture for semantic enrichment and query processing is suggested in Figure 1. Other architectures such as peer-to-peer are also possible. The global query is submitted to the *coordinator*. The coordinator is in charge of resolving ontological differences, and supervising query optimization and execution. Query processing itself can be carried out in many different ways utilizing a combination of sources and the coordinator. We briefly discuss some of these options below.

The simplest approach to query processing in this architecture is for the coordinator to (1) resolve ontological differences with the aid of ontology servers, (2) request each source to materialize their fragments of properties that appear in the query, (3) collect the fragments from sources, (4) perform taxonomic resolution on the data and materialize global properties, and (5) execute the query. Although conceptually simple, this approach can be inefficient in practice.

An alternative approach is for the coordinator to decompose the query and submit the subqueries to the sources for processing. Then collect these results, and possibly execute further queries on these intermediate

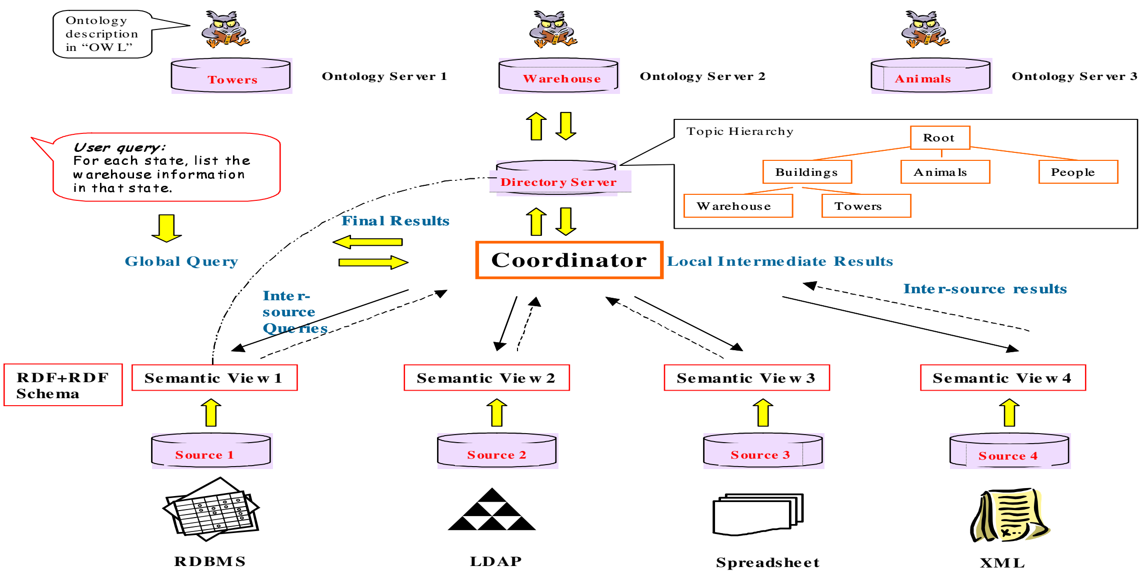

Figure 1: An architecture for semantic markup and query processing.

results to generate the final answer. Interesting optimization opportunities are possible in this alternative plan. We will use a simple example to discuss this approach. More details can be found in [11].

Consider a simple global query that involves the join of two properties p and q, $p \bowtie q$. Suppose there are n sources, $s_1, \ldots, s_n$. The expansion of the query results in the following subqueries: $p_1 \bowtie q_1, \ldots, p_1 \bowtie q_n$, $\ldots p_n \bowtie q_1, \ldots, p_n \bowtie q_n$. Among these subqueries, there are n *local* subqueries $p_1 \bowtie q_1, \ldots, p_n \bowtie q_n$. These subqueries refer only to fragments in a single source and can be processed solely by that source. The rest need fragments from different sources and are called *inter-source* sub-queries. In general, if the query involves k properties and the number of sources is n, there can be up to n local sub-queries and $O(n^k)$ inter-source sub-queries. An important optimization opportunity here is that if certain integrity constraints (key and foreign key constraints) hold then some inter-source sub-queries need not be processed at all. E.g., suppose the functional dependency $q : Y \rightarrow Z$ holds for property q, and according to source s_i, the foreign key constraint (RIC) $p_i[Y] \subseteq q_i[Y]$ holds, where Y is the join argument in $p \bowtie q$. Then, we can show that all inter-source queries involving p_i are redundant. Indeed, we have characterized cases where inter-source sub-queries can be avoided, and have also developed algorithms to derive foreign-key integrity constraints such as above. Local sub-queries can be reformulated in terms of the local source data structure (*e.g.* XML document), and executed (partly or in full) locally. Interested readers are referred to [11] for details. We have also developed optimization techniques for the execution of inter-source sub-queries for cases that require inter-source processing. Note that previous work on distributed query optimization can be leveraged, although the presence of ontologies, availability of semantic knowledge in the form of integrity constraints, and the differences in the data storage formats, mean that the cost model for the optimization should be revisited. More work is needed in the future to address these issues thoroughly. While distributed query optimization is not new, the twists brought about as a result of the Semantic Web are indeed new and challenging.

3 Under the covers: What is needed?

In this section we discuss the question "what is needed to make X-DARES-U approach, or more generally, wide scale interoperability and integration successful?" We propose the following criteria for measuring success: (1) It should be relatively easy for a new source to join a federation of interoperable sources, (2) Query formulation should be simple, (3) Query processing should have an acceptable performance, and (4) Last, but not least, the system should be scalable to large number of sources.

Tools and techniques from several domains are needed for the ultimate success of this project. Some of these are already available, some are at different stages of development, and some still need to be developed. We provide a summary below.

3.1 Semantic Web tools and techniques

The Semantic Web initiative has taken the first important steps in realizing languages and tools needed for interoperability. Resource Description Language (RDF) [16], RDF Vocabulary Description Language (RDF Schema) [17], the DARPA Agent Markup Language (DAML) [5], and OWL Web Ontology Language [14] are examples of languages that are already developed or are under development. They facilitate semantic markups and ontology management. Many companies and research institutes are actively developing tools and systems for these languages (see, for example, [7]).

While ontology description languages are quite expressive, in our opinion, more powerful tools are needed for ontology management, especially for describing relationships among concepts and properties of two or more different independently-designed ontologies (please also see related discussions on ontology languages and ontology maintenance in Section 3.2). There is a fine balance between expressive power of languages and complexity (efficiency) of operations in these languages. The key is to maximize the expressive power while maintaining an acceptable performance. There is also an acute need for efficient *ontology brokers* to reconcile ontological and taxonomic differences among sources in order to make interoperability possible.

3.2 Theoretical and conceptual questions and solutions

Knowledge representation and ontology description and maintenance have been active research areas for decades. The Semantic Web initiative has contributed new languages, such as DAML and OWL, which pay special attention to *efficiency* as well as *expressive power*. Description Logics (DL) provide the mathematical basis for ontology languages. In fact, OWL itself has been inspired and influenced by DLs [1]. When multiple ontologies are used by information sources, there is a need to determine and declare inter-ontology relationships. An expressive language is needed for the declaration of these relationships, and (semi)automatic techniques are needed for their discovery. These topics, sometimes referred to as *ontology integration*, *ontology matching*, and *ontology alignment* are active areas of research [4]. AI research is rich with respect to matching and similarity-based algorithms, and some of these techniques have been applied recently to schema and ontology matching [8, 13]. However, several fundamental questions have yet to be answered. For instance, how are we to choose an appropriate semantic model and mapping for a source? What (technical and other) incentives can we offer a source administrator other than the promise that they can join a "federation"?

Optimization and processing of (global) queries in a multi-source environment poses many challenges, some of which were discussed in Section 2.3. Of particular interest is the case where multiple ontologies are employed. For example, consider a (global) query Q that uses an ontology $\mathcal{O}$, and assume a source i uses a different ontology $\mathcal{O}_i$. The algorithm that generates the local sub-query Q_i at source i needs to perform a *maximal sound rewrite* of Q in terms of the ontology $\mathcal{O}_i$, in the sense that answers to the rewritten query are provably answers to the original query and there is no rewrite which produces a proper superset of such valid answers. The algorithm for and the complexity of this task critically depend on the expressive power of the language used to declare inter-ontology relationships. We have explored this question for a simple language that allows concept/property equivalence as well as sub-concept and sub-property declarations in [11]. Further investigation is needed for more expressive languages.

An orthogonal issue in query optimization is the avoidance or minimization of redundant processing. While some initial work on identifying and inferring types of integrity constraints useful for this form of query optimization has been done in [6, 11], more work is needed on this question.

When overlapping information across sources is consistent, taking their union (with or without duplicate elimination) is just one possible way of combining them. Other operators (e.g., intersection) may be meaningful under some circumstances. When there *is* inconsistency across data sources, how should we resolve them? How can we answer queries reliably and efficiently in this case? Some proposed approaches in database research are based on accociating *certainty* (or *reliability*) factors with data, either directly or indirectly by associating these factors with information sources. In the Semantic Web approach, this can be accomplished elegantly by *reification*. Storage and processing efficiency for reified statements becomes of utmost importance for this type of application that makes very heavy use of reification. The implementors of the Semantic Web framework Jena have addressed this issue in their recent (Jena2) release [21].

There are thus many deep semantic, algorithmic, and complexity questions that provide fertile ground for further research.

3.3 Database tools and techniques

Efficiency is one of the most important goals of an interoperability system that incorporates large number of sources. We need to investigate a multitude of techniques from database research (such as indexing, caching, and optimization using materialized views), as well as develop new techniques related to directory organization and maintenance, ontology broker and server systems, and web-based peer-to-peer systems to harness the complexity of scale. Our work in [11] investigates interesting *semantic* optimization, *i.e.* optimization made possible by data integrity constraints such as key and foreign key constraints.

The *cost-based* query optimization technique has been applied quite successfully in (centralized and distributed) database systems. In this approach, a small number of query execution plans are generated for a given query, and the cost of each plan is estimated. The plan with the lowest estimated cost is then selected for processing the query. We believe cost-based optimization techniques developed for distributed databases should prove useful but will need to be gracefully adapted to deal with the following challenges: (i) The search space is more explosive than for distributed databases which have a small number of sources; (ii) Statistics are harder to collect in a wide distributed context such as ours; (iii) Server loads and network traffic may change dynamically and rapidly. As discussed in Section 2.3, there are many important differences in this application compared to classical distributed query processing that arise from (i) the presence of ontologies, (ii) the availability of semantic knowledge that can be leveraged for substantial query optimization, and (iii) the differences in the data storage formats. We believe large scale (XML) data integration and interoperability will be a killer application for the Semantic Web. For this application to be successful, the aforementioned challenges must be addressed.

3.4 Standards

Any information integration and interoperability system is ultimately dependent on the establishment of standards. It is unrealistic to expect that a single, universally accepted ontology will be established and used by every information source. Even if we restrict ourselves to a single application domain (e.g., purchase order processing), we cannot expect a single ontology that is accepted by everybody. On the other hand, it is unlikely that every source involved in an application will use a different ontology. Instead, community wide ontologies offer a nice hierarchical level in between. Our thesis is that for each application domain a relatively small number of (community wide) ontologies will be established and used by a large percentage of information sources in that domain. Efforts towards this goal are already underway by various groups of parties engaged in specific applications [3]. More efforts for integrating multiple ontologies related to a single application domain are needed and need to be done in an incremental fashion.

4 Conclusions

Interoperability and information integration is a long standing unresolved problem with tremendous application pull. The advent of XML has created a framework akin to low level plumbing in which this problem can be profitably studied. Our vision is wide scale interoperability and integration. The recent Semantic Web initiative with its host of technologies brings several relevant and useful tools and techniques to the table which we can exploit in the interoperability context. However, this raises a variety of challenges and questions at various levels – from fundamental theoretical to systems, to technological to social. In our X-DARES-U project, we are investigating several of these technical challenges. There is a well recognized need for a killer application for the Semantic Web. We argue that wide scale interoperability and integration is a great killer application and invite researchers, developers, and industries to take on that challenge. Tim Berners-Lee [2] advocates publishing and processing everything in RDF using ontologies. We thus believe the timing is just perfect for wide scale interoperability to be tackled, riding this ontology wave!

References

[1] F. Baader, I. Horrocks, and U. Sattler. Description logics as ontology languages for the semantic web. In D. Hutter and W. Stephan, editors, *Festschrift in honor of Jörg Siekmann*, Lecture Notes in Artificial Intelligence. Springer, 2003. To appear.

[2] T. Berners-Lee: Semantic Web: Where to direct our energy. Invited talk. International Semantic Web Conference ISWC'03. `http://iswc2003.semanticweb.org/invitedtalks.html.`

[3] T. Berners-Lee and E. Miller. The semantic web lifts off. *ERCIM News*, 51:9–11, 2002.

[4] D. Calvanese, G. D. Giacomo, and M. Lenzerini. A framework for ontology integration. In *Proceedings of Semantic Web Working Symposium*, 2001.

[5] The DARPA agent markup language (DAML). `http://www.daml.org/`.

[6] S. Davidson, W. Fan, C. Hara, and J. Qin. Propagating XML constraints to relations. In *Proceedings of IEEE International Conference on Data Engineering*, 2003.

[7] M. Denny. Ontology building: A survey of editing tools. `http://www.xml.com/pub/a/2002/11/06/ontologies.html`, 2002.

[8] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proceedings of the International WWW Conference*, pages 662–673, 2002.

[9] A. Halevy, O. Etzioni, A. Doan, Z. Ives, J. Madhaven, L. McDowell, and I. Tatarinov. Crossing the structure chasm. In *Proceedings of Conference on Innovative Data Systems Research*, 2003.

[10] L. V. S. Lakshmanan. Is the Semantic Web a Database? Keynote Talk, *Int. Database Engineering and Applications Symposium*, July 2003, Hong Kong.

[11] L. V. S. Lakshmanan and F. Sadri. Interoperability on XML data. In *Proceedings of the International Semantic Web Conference*, 2003.

[12] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proceedings of International Conference on Very Large Databases*, pages 251–262, 1996.

[13] N. F. Noy and M. A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In *AAAI/IAAI*, pages 450–455, 2000.

[14] OWL web ontology language. `http://www.w3c.org/TR/owl-features`.

[15] L. Popa, Y. Velegrakis, R. J. Miller, M. A. Hernández, and R. Fagin. Translating web data. In *Proceedings of International Conference on Very Large Databases*, 2002.

[16] Resource Description Framework (RDF). `http://www.w3c.org/RDF/`.

[17] RDF vocabulary description language (RDF Schema). `http://www.w3c.org/TR/RDF-schema/`.

[18] Semantic Web. `http://www.w3c.org/2001/sw/`.

[19] Amit Sheth and James Larson. Federated database systems for managing distributed heterogeneous, and autonomous databases. ACM Computing Surveys, 22(3):183–236, September 1990.

[20] The TSIMMIS project home page. `http://www-db.stanford.edu/tsimmis/tsimmis.html`.

[21] K. Wilkinson, C. Sayers, H. Kuno, and D. Reynolds. Efficient RDF storage and retrieval in Jena2. In *Proceedings of VLDB Workshop on Semantic Web and Databases*, pages 131–150, 2003.