ISENS: A System for Information Integration, Exploration, and Querying of Multi-Ontology Data Sources

Dimitre A. Dimitrov, Roopa Pundaleeka
Tech-X Corp.
Boulder, CO 80303, USA
Email: {dad, roopa}@txcorp.com

Abir Qasem, Jeff Heflin
Lehigh University
Bethlehem, PA 18015, USA
Email: {abq2, heflin}@cse.lehigh.edu

***Abstract*—Separate data sources on related domains of knowledge generally contain different but complementary information. There are queries that can only be answered with pieces of information from some of the separate sources. Thus, it is of considerable interest to enable query answering based on searching the information in an integrated collection of sources. However, independently developed and evolved data sources generally use different schemas to represent their data. This makes it difficult to search the sources in an integrated way. To address this problem, we have developed an end-to-end information integration system that is based on the Semantic Web technologies, algorithms for efficient source selection, and a Web-based user interface to construct queries and search multi-ontology data. We describe the system architecture, handling of information integration in a multi-ontology environment, and the user interface capabilities for ontology visualization, query construction, and presentation of results. Moreover, we have developed a multi-ontology real-world data set up and measured the performance of the different ISENS components on query answering that involves information integration from the different data sources. The results indicate that the source selection and the logical reasoning parts of query processing are dominated by the time to transfer data from the selected sources and to load the triples in the reasoner. However, the query answering time is often of the order of a second or less allowing ISENS to be used efficiently for such semantic applications.**

***Keywords*-Semantic Web; information integration; distributed query answering**

I. Introduction

Different organizations often develop data management solutions for related domains of knowledge, but independently design heterogeneous schemas to describe their data. Moreover, the data is usually semi-structured, contains binary entities (e.g. images), and is distributed - available from different data servers on a network.

Finding relevant information from all available and diverse data sources requires solving a number of problems related to data representation and information integration in a distributed environment [1]. We describe here the end-to-end ISENS system that we designed and implemented to address these problems. We integrated and extended Semantic Web [2] technologies to handle information integration (based on logical views [3]) and querying of distributed data.

To encode data and address its representation we used the Resource Description Framework (RDF) and the Web Ontology Language (OWL). For query representation, we use the SPARQL[1] language. It is specifically designed for the Semantic Web and enables constructing sophisticated queries to search for different types of data. For information integration over data described by multiple OWL ontologies, we implemented a solution based on map ontologies, source descriptions, and source selection algorithms that we developed for efficient identification of relevant Semantic Web data sources during query answering [4], [5], [6], [7].

Furthermore, we specifically investigate how the ISENS system performs when applied to information integration tasks using real-world semantic data. Initially, the algorithms we implemented were tested on synthetically generated data [4], [6].

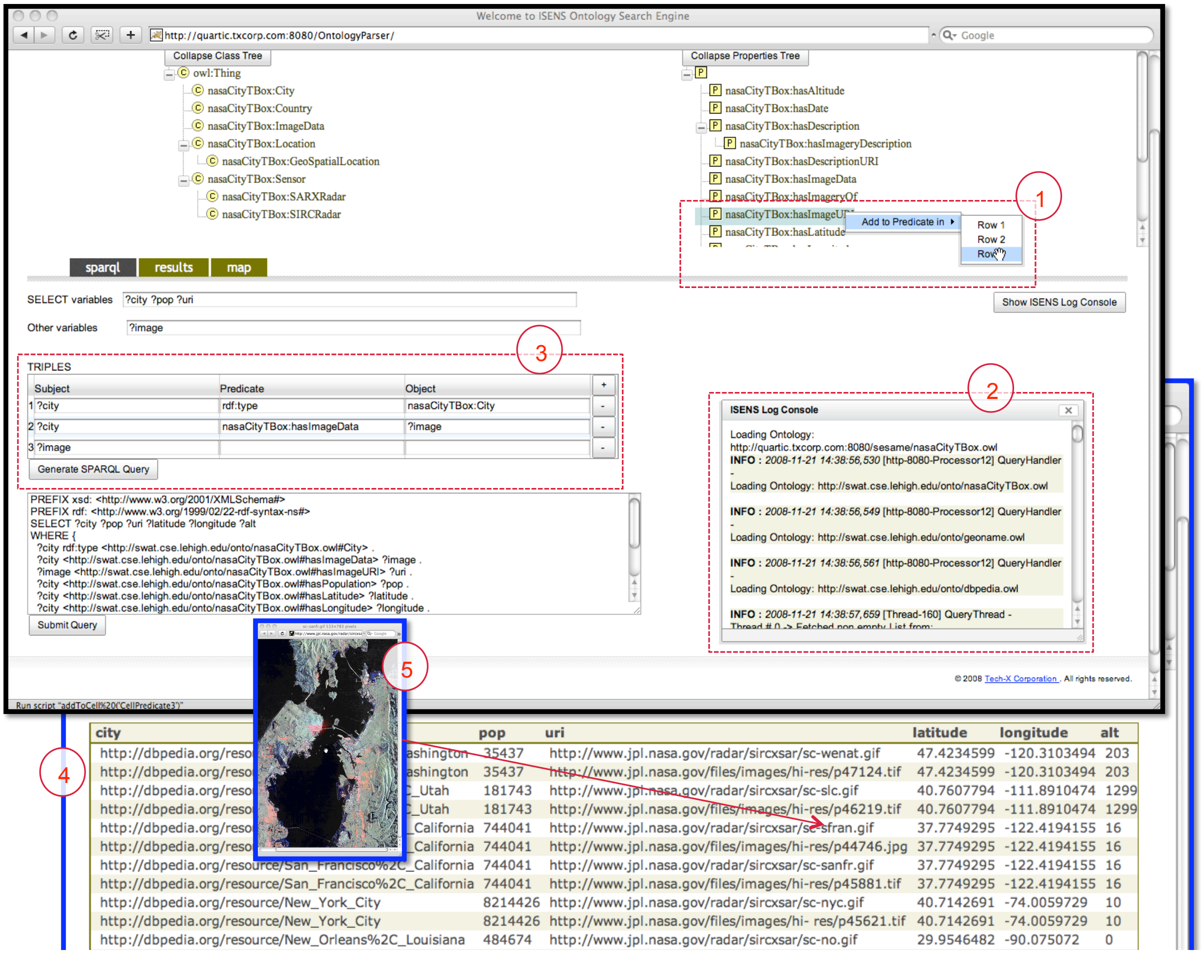
We incorporated different real-world data sources that commit to different ontologies into the ISENS system. Then, we created specific map ontologies and source descriptions. Given this setup, ISENS can answer queries that require information to be combined from the different sources. Such queries cannot be answered by any of the three sources independently.

The rest of the paper is organized as follows. In Section II, we discuss the ISENS system components and describe specific capabilities of the Web-enabled graphical user interface. We describe the real-world data sources we used, map ontologies, and source descriptions in Section III. Next, we explain the information integration capabilities and present results from performance measurements on query answering.

II. ISENS User Interface and Query Processing Components

ISENS consists of three main components: the Ontology Based Information Integrator (OBII), the Web-based graphical user interface (GUI), and a set of OWL ontologies, data sources with ontology individuals, map ontologies, and relevance meta data. Since we have described the ISENS

[1] SPARQL query language for RDF, W3C Recommendation 2008, http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/



SELECT variables ?city ?pop ?uri

Other variables ?image

TRIPLES

	Subject	Predicate	Object
1	?city	rdf:type	nasaCityTBox:City
2	?city	nasaCityTBox:hasImageData	?image
3	?image		

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?city ?pop ?uri ?latitude ?longitude ?alt
WHERE {
  ?city rdf:type <http://swat.cse.lehigh.edu/onto/nasaCityTBox.owl#City> .
  ?city <http://swat.cse.lehigh.edu/onto/nasaCityTBox.owl#hasImageData> ?image .
  ?image <http://swat.cse.lehigh.edu/onto/nasaCityTBox.owl#hasImageURI> ?uri .
  ?city <http://swat.cse.lehigh.edu/onto/nasaCityTBox.owl#hasPopulation> ?pop .
  ?city <http://swat.cse.lehigh.edu/onto/nasaCityTBox.owl#hasLatitude> ?latitude .
  ?city <http://swat.cse.lehigh.edu/onto/nasaCityTBox.owl#hasLongitude> ?longitude .
```

```
Loading Ontology:
http://quartic.txcorp.com:8080/sesame/nasaCityTBox.owl
INFO : 2008-11-21 14:38:56,530 [http-8080-Processor12] QueryHandler
-
Loading Ontology: http://swat.cse.lehigh.edu/onto/nasaCityTBox.owl

INFO : 2008-11-21 14:38:56,549 [http-8080-Processor12] QueryHandler
-
Loading Ontology: http://swat.cse.lehigh.edu/onto/geoname.owl

INFO : 2008-11-21 14:38:56,561 [http-8080-Processor12] QueryHandler
-
Loading Ontology: http://swat.cse.lehigh.edu/onto/dbpedia.owl

INFO : 2008-11-21 14:38:57,659 [Thread-160] QueryThread -
Thread # 0 -> Fetched non empty List from:
```

city	pop	uri	latitude	longitude	alt
http://dbpedia.org/reso ashington	35437	http://www.jpl.nasa.gov/radar/sircxsar/sc-wenat.gif	47.4234599	-120.3103494	203
http://dbpedia.org/reso ashington	35437	http://www.jpl.nasa.gov/files/images/hi-res/p47124.tif	47.4234599	-120.3103494	203
http://dbpedia.org/reso C_Utah	181743	http://www.jpl.nasa.gov/radar/sircxsar/sc-slc.gif	40.7607794	-111.8910474	1299
http://dbpedia.org/reso C_Utah	181743	http://www.jpl.nasa.gov/files/images/hi-res/p46219.tif	40.7607794	-111.8910474	1299
http://dbpedia.org/reso _California	744041	http://www.jpl.nasa.gov/radar/sircxsar/sc-sfran.gif	37.7749295	-122.4194155	16
http://dbpedia.org/reso _California	744041	http://www.jpl.nasa.gov/files/images/hi-res/p44746.jpg	37.7749295	-122.4194155	16
http://dbpedia.org/resource/San_Francisco%2C_California	744041	http://www.jpl.nasa.gov/radar/sircxsar/sc-sanfr.gif	37.7749295	-122.4194155	16
http://dbpedia.org/resource/San_Francisco%2C_California	744041	http://www.jpl.nasa.gov/files/images/hi-res/p45881.tif	37.7749295	-122.4194155	16
http://dbpedia.org/resource/New_York_City	8214426	http://www.jpl.nasa.gov/radar/sircxsar/sc-nyc.gif	40.7142691	-74.0059729	10
http://dbpedia.org/resource/New_York_City	8214426	http://www.jpl.nasa.gov/files/images/hi- res/p45621.tif	40.7142691	-74.0059729	10
http://dbpedia.org/resource/New_Orleans%2C_Louisiana	484674	http://www.jpl.nasa.gov/radar/sircxsar/sc-no.gif	29.9546482	-90.075072	0

Figure 1. Example Web browser windows from ISENS demonstrate interactive capabilities for ontology visualization, efficient construction of SPARQL queries, and presentation of results.

back-end components previously [4], [6], we mention here only essential information about them and cover in more detail the capabilities of the ISENS user interface.

The ISENS GUI is Web-based and can be accessed via modern Web browsers. It provides easy-to-use functionality typically available from stand-alone applications. The ISENS GUI has dynamically changing and asynchronous capabilities similar to modern JavaScript, AJAX-based, Web applications. Example Web browser window captures of the ISENS GUI are shown in Fig. 1.

One of our main goals with the ISENS GUI is to enable users to visually explore taxonomies of OWL ontologies and to have easy-to-use capabilities to construct SPARQL queries using terms from loaded ontologies. ISENS allows any OWL ontology available on the Web to be loaded and its taxonomy visualized using the ISENS GUI. This capability is shown in the top part of Fig. 1 for one of the ontologies we created to describe a specific data source.

Creating an easy-to-use GUI to enable users to construct SPARQL queries is a challenging problem. The ISENS GUI enables users to create SPARQL queries more easily and efficiently in two ways. A term from a visualized ontology can be selected and then can be assigned (via mouse input) to a specific field of a chosen SPARQL query triple. Box #1 in Fig. 1 shows this capability. It demonstrates how a property can be selected and assigned to one of the predicates of the three instantiated SPARQL query triples (in box #3). ISENS is aware that ontology property names can be assigned only to the predicate fields of triples while the ontology class names can go to either the subject or object fields. Secondly, users can start typing directly in a triple field and ISENS will automatically open a window to prompt with all allowed ontology terms that start with the typed string. The window will include terms from all ontologies a user has previously loaded. ISENS caches loaded ontologies for efficiency but also allows users to reload ontologies and thus update their state in the cache.

These two capabilities enable users to save time by not having to type long names for ontology terms that are usually dominated by name space strings. The relevant name space strings are automatically generated by ISENS, together with appropriate PREFIX strings. Once a user creates specific triples, ISENS provides automatic generation of a SPARQL query that consists of the triple and the user specified variables. Users can designate variables as selected (for the answer set) and others. The latter can be used only in the body of a query. The query appears in the text area shown in Fig. 1 below the triples. If needed, this area allows further editing and refinement of the auto-generated SPARQL query before submitting for answering. Once a query is submitted, its processing progress can be monitored in the ISENS Log Console shown in box #2 of Fig. 1. This console is also useful for debugging. Query results are presented in a separate tab using a table view (shown partly in box #4). All variable substitutions in an answer that are URLs are automatically made active and open in separate windows (as in box #5) if clicked on. We have developed the ISENS GUI using JavaScript and functionality from the Yahoo! User Interface Library[2] (YUI).

The query answering is handled by the OBII component. It does the information integration from different data sources that (generally) commit to different ontologies. OBII relies on OWL maps and relevance (REL) meta data statements, together with the defined ontologies the data are described in, to perform its task. We explain in detail ontology maps and REL statements in Subsection III-B. In OBII, we have implemented two source selection algorithms—the goal node search [6] (GNS) algorithm and our original source selection algorithm [4] (we adapted the peer data management system (PDMS) reformulation algorithm [8] to the Semantic Web) in this module. PDMS extends the standard Local-as-View (LAV) and Global-as-View (GAV) information integration approaches [9]. These algorithms determine the URLs for the potentially relevant sources from which to request data to answer a query. Once the data from the selected sources is obtained, OBII uses the **KAON2**[3] logical reasoner to find the answers. OBII is written in Java.

Due to the limited space here, the description of the theoretical underpinning of our work could not be included. We refer interested readers to Qasem et al.[4] for a formal description of the source selection framework and Qasem et al. [6] for an extended exposition of our algorithms.

III. Experiments with Real-World Data

Here, we discuss first the specific real-world data sources we selected and prepared for testing the ISENS system. Then, we mention issues involved in setting up data to enable query answering. Finally, we discuss query testing and performance measurement results.

A. Data Sources

We harvested and prepared three real-world data sources: DBpedia[4], Geonames[5], and imagery-related information on cities from NASA[6]. After translating the data to OWL, we loaded the OWL-formatted data into a Sesame[7] knowledge base. The different data sources are loaded into different Sesame stores. These data sources provide complementary information on a similar domain of knowledge allowing to answer queries that require data from the different sources once they are integrated.

We selected about $33,000$ cities from DBpedia and decided to use only the following set of properties that cover a large number of individuals: population, population as of, postal code, image map, established date and airport code. However, in order to experiment with map composition (described in Subsection III-B) to answer specific queries, we extended the DBpedia ontology with two more properties: `db:latitude` and `db:longitude` (though there are no DBpedia ontology individuals marked up with these two properties).

The Geonames contains over eight million geographical names and consists of 6.5 million unique features of 2.2 million populated places. We collected city specific data of about $70,000$ cities from this source. The properties of cities we collected from Geonames are latitude, longitude, elevation and country code. The Geonames ontology individuals are the only ones that contain information about latitude and longitude. The Geonames is the largest data set of the three at approximately 27 megabytes (MB).

The third data source consists of NASA provided imagery and related semi-structured textual data on cities. The NASA data are in HTML format. We designed and developed an OWL ontology to describe the NASA city data. Then, to generate ontology individuals, we wrote code to crawl the NASA data, extract relevant information from the fetched HTML pages, and generate OWL individuals in the terms of the developed ontology. The code generates individuals about the 33 cities for which imagery was provided.

The ISENS data setup consists of domain ontologies in OWL, data individuals (stored in Sesame) that commit to the domain ontologies, map ontologies (connecting terms among the three domain ontologies), and relevance statements. This data configuration is also relevant for using Semantic Web technologies within “Deep Web” settings [10].

B. Map Ontologies and Source Relevance Meta Data

We use map ontologies in order to align information from heterogeneous ontologies. In our solution, we consider OWL axioms (a subset of OWL to be precise) to map terms in

[2] Yahoo! User Interface Library: http://developer.yahoo.com/yui/

[3] http://kaon2.semanticweb.org/

[4] http://dbpedia.org/

[5] http://www.geonames.org/

[6] http://www.jpl.nasa.gov/radar/sircxsar/cities.html

[7] Aduna: Sesame RDF Framework, http://www.openrdf.org

domain ontologies. We have developed a mapping language for OWL information integration (OWLII) [4], [7] to specify the map ontologies.

The OWLII mapping language is more expressive than Description Horn Logic. It is a subset of OWL DL (thus decidable) and is compatible with information integration algorithms that use both LAV and GAV rules. The support for LAV and GAV rules is needed in our adaptation of the PDMS algorithm for the Semantic Web.

Apart from simple equivalences and subsumption that address the bulk of heterogeneity in Semantic Web ontologies, OBII also enables to map some interesting (and quite common) forms of heterogeneity. Qasem *et al.* [4], [7] give a comprehensive description of the OWLII mapping language.

ISENS also supports map composition. Thus, if there is a map ontology M_{12} between domain ontologies O_1 and O_2 and another map M_{23} between domain ontologies O_2 and O_3, ISENS can traverse these maps effectively supporting map composition between the O_1 and O_3 domain ontologies. As an example of map composition, consider the pair of properties `nasa:hasLatitude` and `nasa:hasLongitude` in the NASA's ontology and the `geo:latitude` and `geo:longitude` pair in the Geonames ontology. We did not include mapping axioms about these properties in the direct map in order to test the map composition support. Instead, we created a pair of properties `db:latitude` and `db:longitude` in the DBpedia domain ontology. Then, we specified mapping axioms in the NASA-DBpedia map ontology to map the NASA's `nasa:hasLatitude` and `nasa:hasLongitude` properties to the corresponding `db:latitude` and `db:longitude` ones in the DBpedia ontology. We also specified mapping axioms in the DBpedia-Geonames map ontology to map DBpedia's `db:latitude` and `db:longitude` properties to the corresponding `geo:latitude` and `geo:longitude` pair of properties in the Geonames ontology. Given these maps, ISENS can effectively compose a map between NASA's `nasa:hasLatitude` and `nasa:hasLongitude` properties and the Geoname's `geo:latitude` and `geo:longitude` ones. Given the support for map composition in ISENS, we could completely remove the NASA-Geonames map and still be able to answer the queries that relied on its map axioms, provided they all can be deduced from axioms in the NASA-DBpedia and DBpedia-Geonames maps.

Apart from ontology maps, a way to specify index-level meta data about sources is needed for efficient querying of large data sets. If we can effectively determine which sources are potentially relevant by using meta data about the sources, then we can limit our queries only to these sources. We use the concept of a source relevance to summarize the content of a data source. ISENS can use this information to select the needed sources to answer a given query. Source relevance (specified in RDF) allows the content of a data source to be declaratively described without regards to a specific access mechanism. We refer to the RDF statements that express a source's relevance to a query as REL statements. ISENS handles the semantic heterogeneity between data sources using the adapted information integration algorithm and the mappings expressed in OWL to reformulate the query into the ontologies that the data sources use. REL statements are formed using a subset of OWL and can be translated into LAV rules. Qasem *et al.* [5], [7] describe REL statements in detail.

C. Query Testing and Performance

Given the three sources of real-world data, a user might be interested to find all the cities for which there are image data available, their populations, latitudes, and longitudes. Such a query can be expressed in a straightforward way in SPARQL. For testing the system, we have written this query[8] in terms of the NASA ontology.

```
SELECT ?city ?pop ?uri ?lat ?lon
WHERE {
  ?city nasa:hasImageData ?imgd .
  ?imgd nasa:hasImageURI ?uri .
  ?city nasa:hasPopulation ?pop .
  ?city nasa:hasLatitude ?lat .
  ?city nasa:hasLongitude ?lon }
```

This query cannot be answered completely by any of the three separate data source by itself.

The NASA data source has ontology individuals about cities with related image data, including image URIs. However, this data source does not have any individuals with information on population, altitude, longitude, and latitude. The prepared DBpedia data source contains individuals on cities and their populations but does not have individuals that can answer any of the predicates of the query that are not on city population. Finally, the Geonames data source is the only one that contains individuals with information on altitude, latitude, and longitude of cities.

However, ISENS can answer this query using its ontology maps to deduce that `hasPopulation` in the NASA ontology is equivalent to the `population` property in the DBpedia one. Finding the relevant latitude and longitude answers is more involved and relies on map composition. As we discussed previously, we setup the ontology maps in such a way that we did not provide direct map axioms to indicate equivalence of latitude and longitude properties from the NASA ontology to the corresponding properties in the Geonames ontology (since the Geonames data source is the only one with individuals on city locations). However, we provided map axioms between the NASA and DBpedia ontologies on these properties and between the DBpedia and

[8] The nasa prefix is defined via:
PREFIX nasa: ¡http://swat.cse.lehigh.edu/onto/nasaCityTBox.owl#¿

the Geonames ones. Testing ISENS on this query, returns the correct values of the latitude and longitude for the cities that satisfy the rest of the query predicates.

After the map ontologies have been loaded and processed, ISENS uses REL statements to find relevant data sources on latitude, longitude data (from the Geonames individuals), population data (from the DBpedia individuals), and image URIs (from the NASA individuals) to answer this query. It then queries each of these sources specifically for the relevant predicates contained within.

Next, we consider the performance of ISENS on query answering using the real-world data setup. We prepared a set of nine queries to test how ISENS performs under different data source requirements. The configuration for the measurements consists of all data sources accessible from a Sesame server on a Tech-X computer with two dual core *AMD Opteron* processors at 2 GHz with 4 MB of total RAM. Each core has 1 MB of L2 cache. The tests were executed using Java 1.6.0 over 2.6.25 64-bit SMP Linux kernel. The Sesame server was configured to use multiple PostgreSQL databases for internal storage of OWL ontology individuals from the different data sources. The query answering performance was measured on the same computer on which the Sesame server was running. We also performed measurements when all data sources are remotely located on a computer at Lehigh University (while still running the tests on the same machine at Tech-X) and verified that the difference in the results is due only to the difference in time to fetch the data over the network relative to the time it takes to get the data from the local (Tech-X) Sesame server. We ran each of the test queries 10 times and report results here after averaging the times from the individual runs. The results from the performance measurements on answering a set of 9 test queries are shown in Table I.

We separate the query answering process in three parts. The "Find Sources" part consists of receiving the SPARQL query, loading the ontology maps, relevance statements, and executing the source selection algorithm to determine the potentially relevant Sesame data stores from which to request ontology triples. This part also generates separate queries for each of the triples that will be requested from the Sesame stores. These queries represent input for threads that are then created and dispatched to fetch the data from the Sesame stores. We have implemented in ISENS the execution of data transfers from selected data source in parallel using Java threads. Note that for most of these queries, the "Find Sources" time taken by the source selection algorithm is considerably smaller compared to the second part of the query answering process—the "Load Sources".

This part consist of creating the threads, running them to transfer the data from the Sesame stores, converting the transferred data to KAON2 triples, and then loading the triples in the KAON2 reasoner. The measurements show that the "Load Sources" part dominates the overall query answering times (also shown in Table I under the "Query Time" column). In particular, the second query (this is the example query we presented at the beginning of this Subsection) takes approximately 15 seconds to complete. This is due to the largest size (36 MB) of the data this query requests as shown in Table I. Moreover, the detailed separation of the "Load Sources" time, given in Table II, into the time to transfer the data from the Sesame stores (join the threads) and then to convert to KAON2 triples and load them into the KAON2 reasoner shows that the conversion together with the loading takes similar amount of time as the data transfer time. Note that query 2, 3, and 9 request the most data. These queries involve data requests from the largest data source, the Geonames. Query 1 requires data from the Nasa and DBpedia sources. Data from the three sources are required to answer query 2. Each the other 7 queries requires data from only one of the three sources with two of them measuring the performance of additionally selecting parts of the answers using SPARQL FILTER functions.

Table I
ISENS PERFORMANCE ON ANSWERING A SET OF QUERIES.

Q	Find Sources (sec)	Load Sources (sec)	KAON2 Time (sec)	Query Time (sec)	Data Size (MB)	Load Bandw. (MB/sec)
1	0.241	0.621	0.057	0.918	1.570	2.529
2	0.644	14.974	0.052	15.670	36.145	2.414
3	0.093	4.457	0.247	4.797	4.750	1.066
4	0.061	0.048	0.019	0.129	0.042	0.859
5	0.053	0.063	0.024	0.140	0.205	3.282
6	0.024	0.625	0.081	0.730	1.941	3.103
7	0.024	0.468	0.062	0.554	1.528	3.268
8	0.027	0.588	0.052	0.667	1.528	2.599
9	0.091	3.508	0.140	3.739	4.750	1.354

Table II
LOAD SOURCE PERFORMANCE FOR THE QUERIES AS IN TABLE I.

Q	Threaded Data Transfer (sec)	Conversion to KAON2 Triples (sec)	Add Triples to KAON2 (sec)
1	0.420	0.028	0.107
2	7.065	0.105	6.818
3	3.945	0.035	0.380
4	0.039	0.0004	0.0011
5	0.045	0.0004	0.0006
6	0.398	0.082	0.145
7	0.323	0.039	0.105
8	0.428	0.041	0.118
9	3.010	0.084	0.366

The "KAON Reasoning" is the third part of the query answering process. It consists of issuing the original SPARQL query to the KAON2 reasoner (after all data has been loaded) and obtaining the set of *complete* answers from it. For the 9 test queries, KAON2 returns 28, 28, 23713, 59, 52, 9597, 7926, 1209, and 2586 answers respectively. The performance results using this real-world data set, as well as from experiments with synthetic data we have done previously [4], show that the reasoning time is not of significant importance. The reasoning always completes for a fraction of a second and takes time that is small relative to the "Load Sources" time.

The performance data indicate that the highest improvement in reducing the overall query answering time can be achieved by further reducing the "Load Sources" time. The parallelization of fetching the data already achieved an important reduction of this time. However, we see additional ways to further reduce it. In particular, ISENS currently issues separate queries to Sesame stores for each of the triple predicates determined by the source selection algorithm. This approach is likely requesting more data than needed. When data are requested for more than one predicate, it is possible to request specific data individuals multiple times since the data for the different predicates are requested independently. If these predicates have overlapping arguments, then their joint will constrain the requested data and thus reduce its size. This will lead to a smaller data size transfer and will additionally save time by loading a smaller data size into KAON2. Moreover, ISENS users could significantly reduce the answering time by designing queries with the most restrictive predicates in order to achieve the most effective reduction of data size requested from the sources. If a query has redundant predicates (or is more general than necessary to obtain a specific answer), ISENS could potentially request more data from the selected sources to satisfy the query predicates and thus resulting in a longer query answering time. Finally, if completeness of the result set is relaxed, additional improvements in efficiency are possible.

IV. Summary and Future Development

The end-to-end ISENS system prototype we developed demonstrates information integration of Semantic Web data from multiple sources that generally commit to different domain ontologies. ISENS relies on ontology maps and relevance statements to integrate information from independently developed and evolved data sources that generally use different schemas to represent their data. Moreover, ISENS can compose mapping axioms using existing maps to map terms from two ontologies event hough there may not be a direct map between these ontologies.

To solve this multi-ontology and multi-data source problem, we developed two algorithms for efficient source selection. The ISENS system also provides a Web-based user interface with capabilities to visualize ontology taxonomies and an easy-to-use way to construct SPARQL queries.

Some of the features we are planning for future development include extending the capabilities of the user interface to allow for presentation of results that include location information, creation of more complex SPARQL queries, and further optimizations to speed up the performance in answering queries. Our approach also handles the instance coreference problem in a scalable way—it identifies ontology individuals that come from different sources but denote the same entity [11].

Acknowledgments

We are grateful to the US DoE for funding this work under the DE-FG02-05ER84171 SBIR grant.

References

[1] A. Y. Halevy, N. Ashish, D. Bitton, M. Carey, D. Draper, J. Pollock, A. Rosenthal, and V. Sikka, "Enterprise information integration: successes, challenges and controversies," in *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM Press, 2005, pp. 778–787.

[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Sci. Am.*, 2001.

[3] J. D. Ullman, "Information integration using logical views," in *Proc. of ICDT*, Delphi, Greece, 1997, pp. 19–40.

[4] A. Qasem, D. A. Dimitrov, and J. Heflin, "Efficient selection and integration of data sources for answering semantic web queries," in *Proceedings of the Second IEEE International Conference on Semantic Computing*. IEEE Comp. Soc. Press, 2008, pp. 245–252.

[5] ——, "An efficient and complete distributed query answering system for semantic web data," Lehigh University, Tech. Rep. LU-CSE-07-007, 2007.

[6] ——, "Goal Node Search for Semantic Web Source Selection," in *WI 08: Proceedings of the International Conference on Web Intelligence*. IEEE Comp. Soc. Press, 2008, pp. 566–9.

[7] ——, "Goal node search for semantic web source selection," Lehigh University, Tech. Rep. LU-CSE-08-010, 2008.

[8] A. Halevy, Z. Ives, D. Suciu, and I. Tatarinov, "Schema mediation in peer data management systems," in *Proc. of ICDE*, 2003. [Online]. Available: citeseer.ist.psu.edu/halevy03schema.html

[9] M. Lenzerini, "Data integration: a theoretical perspective," in *PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. New York, NY, USA: ACM, 2002, pp. 233–246.

[10] A. Qasem, D. A. Dimitrov, and J. Heflin, "ISENS: A Multi-ontology Query System for the Semantic Deep Web," in *Proc. of the Workshop on the Semantic Web Meets the Deep Web, IEEE CEC08 and EEE08*. IEEE Comp. Soc. Press, 2008, pp. 396–9.

[11] ——, "Towards scalable information integration with instance coreferences," in *Submitted to Workshop on Information Integration on the Web, IJCAI*, 2009.