Knowledge Sifter: Agent-Based Ontology-Driven Search over Heterogeneous Databases using Semantic Web Services

Larry Kerschberg, Mizan Chowdhury, Alberto Damiano, Hanjo Jeong, Scott Mitchell, Jingwei Si, and Stephen Smith

E-Center for E-Business, George Mason University, Fairfax, VA, USA
{kersch, mchowdh1, adamiano, hjeong, smitche2, jsi, ssmith7 } @ gmu.edu
http://eceb.gmu.edu

Abstract. Knowledge Sifter is a scaleable agent-based system that supports access to heterogeneous information sources such as the Web, open-source repositories, XML-databases and the emerging Semantic Web. User query specification is supported by a user agent that accesses multiple ontologies using an integrated conceptual model expressed in the Web Ontology Language (OWL). A collection of cooperating agents supports interactive query specification and refinement, query decomposition, query processing, as well as result ranking and presentation. The Knowledge Sifter architecture is general and modular so that ontologies and information sources can be easily incorporated. A proof-of-concept implementation shows how Knowledge Sifter can search geo-spatial ontology services such as the USGS Geographic Names Information System (GNIS) and Princeton University's WordNet as well as image databases including Lycos and TerraServer. Each Agent is implemented as a Web Service and the external sources are also accessed via Web Service Technology.

1 Introduction

One important problem faced by the Scientific Database Community is the *integration* of data and knowledge from multiple heterogeneous sources. Examples can be found in diverse application domains such as Earth Science, Bioinformatics, and Space Science. The federated approach [18, 28, 29, 31, 49, 52, 53] is used to allow scientists to maintain control of their data, while sharing it within the community. Many of the sharing protocols are ad hoc, and our goal is to provide a framework and architecture by which search and sharing can be easily implemented using standard protocols.

The Knowledge Sifter project, underway at George Mason University, has as its primary goals: 1) to allow users to perform ontology-guided semantic searches for relevant information, both organic and open-source, 2) to refine searches based on user feedback, and 3) to access heterogeneous data sources via agent-based knowledge services. Increasingly, users seek information outside of their own

communities to open sources such as the Web, XML-databases, and the emerging Semantic Web.

The Knowledge Sifter project also wishes to use open standards for both ontology construction and for searching heterogeneous data sources. For this reason we have chosen to implement our specifications and data interchange using the Web Ontology Language (OWL), and Web Services for communication among agents and information sources.

This paper presents the Knowledge Sifter framework as applied to searching and ranking image data from several sources.

2 Related Work

2.1 Semantic Search

Current search technology is keyword-based, while many users may prefer to formulate queries in terms of high-level semantic concepts that are more in tune with standard nomenclature and his tacit knowledge. Moreover, current search engines use proprietary indexing and rating algorithms, so one cannot analyze the search result rankings. In many cases the results returned by search engines are completely irrelevant. In our research on WebSifter, we have identified a *semantic gap* [32, 33] between the way users think and conceptualize a problem, and the primitive way they must pose queries to search engines. We have published our search-result ranking algorithms, introduced learning mechanisms to evolve user preferences, and have demonstrated these concepts in the WebSifter prototype system [34].

Some research incorporates user domain knowledge for semantic search. For example, Aridor et al. [2] represent user domain knowledge as a small set of example web pages provided by users. Chakrabarti et al. adopted both a pre-defined (but modifiable) taxonomy and a set of example user-provided web pages as domain knowledge [6]. OntoBroker [8, 12, 13] uses an ontology in its search. The recent work on WebSifter [32-35], shows that the use of domain knowledge, agent-based services, and personalized ranking metrics can improve both precision and recall. Cercone et al discuss the use of machine translation, machine learning, and user interface design in intelligent search services [5].

2.2 Ontology Specification

An ontology is an explicit specification of a conceptualization. In an intelligent agent system, the ontology is a declarative formalism, the vocabulary for the representation of knowledge for a specific domain. Ontology definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. [16, 17]

The Knowledge Sifter approach posits a conceptual domain model, used in conjunction with a collection of light-weight and specialized ontologies that can be accessed by the user, or his agents, in formulating queries and more complex

scenarios. Some ontologies are domain-independent, e.g., temporal/spatial concepts, while others are domain-dependent, e.g., an image metadata ontology such as the ISO 19115/19139 standard [26, 27].

We envision that the various scientific communities will continue to invest resources in creating such specialized ontologies using domain specific markup languages such as Microarray Markup Language (MAML) for gene expression databases, Protein Extensible Markup Language (PROXIML), Geography Markup Language (GML), and the Intelligence Community Markup Language [25].

Many of the new markup languages will incorporate type concepts such as RDF [9, 22, 36, 37] and XML Schema and there is growing acceptance Semantic Web [3, 19, 20] constructs such as DAML [1, 10, 21], OWL [11].

Organizational ontologies can be complemented with personal ontologies created by users over the course of their investigations. These reflect personal preferences regarding: 1) how concepts are related and organized, 2) preferred search engines, and 3) opinions regarding the authoritativeness of a source and the accuracy of its information. These preferences can be used to rank, sift and winnow the results returned from the heterogeneous data/knowledge sources.

Recently, the issue of creating an ontology for GRID environments has been proposed [51]. The authors point out that "GRID environments are service-oriented and emphasize operations that can be performed on data using associated metadata schemas, rather than focusing upon the content of metadata schemas and relationships between schema elements." In order for middleware services to be able to respond to service requests, the metadata (both schema and instance levels) must be available to that service. The metaphor for Knowledge Sifter is that each query has an associated instantiation which is exchanged by the agents and updated based on the services performed. This captures the pedigree and provenance of the entire collection of activities associated with the instance.

2.3 Semantic Web Services

Web services provide a means for computers and agents to discover, configure, invoke and use specialized programs that have been appropriately "wrapped" in the Web services protocols: Universal Description, Discovery and Integration (UDDI) [47], Simple Object Access Protocol (SOAP) [56], and Web Services Description Language (WSDL) [7].

Part of the DAML+OIL [14] effort is DAML-S, a trio of extensions to DAML designed to automate the specification and advertisement of semantic web services. DAML-S consists of a *Service Profile Ontology* for advertising the capabilities and requirements of a web service; a *Service Model Ontology* to describe how a service operates; and a set of *Service Groundings* which specify how an atomic web service is accessed. The emerging Semantic Web Services [40, 48] motivate the research for Knowledge Sifter Web services.

3 The Knowledge Sifter Agent Architecture

The rationale for using agents to implement intelligent search and retrieval systems is that agents can be viewed as autonomous and proactive. Each agent is endowed with certain responsibilities and communicates using an Agent Communication Language [15]. Recently, Huhns [24] has noted that agents can be thought of a Web services, and this is the approach we have taken to implement the agent community comprising Knowledge Sifter. The family of agents presented here is a subset of those incorporated into the large vision for Knowledge Sifter. This work is motivated by earlier research into Knowledge Rovers [29, 30] performed at GMU. Note that the Knowledge Sifter architecture is quite general, as is its implementation to be discussed in section 5.

The Knowledge Sifter conceptual architecture is depicted in Figure 1. The architecture has three layers: User Layer, Knowledge Management Layer and Data Layer. Specialized agents reside at the various layers and perform well-defined functions. This collection of cooperating agents supports interactive query specification and refinement, query decomposition, query processing, integration, as well as result ranking and presentation. The Knowledge Sifter architecture is general and modular so that new ontologies and new information resources can be easily incorporated.

These services are described below.

3.1 User and Preferences Agents

The User Agent interacts with the user to elicit user preferences that are managed by the Preferences Agent. These preferences include the relative importance attributed to terms used to pose queries, the perceived authoritativeness of Web search engine results, and other preferences to be used by the Integration Agent. The Preferences Agent can also learn the user's preference based on experience and feedback related to previous queries.

3.2 Ontology Agent

The Ontology Agent accesses an imagery domain model which is specified in the Web Ontology Language (OWL). In addition, there are two authoritative name services: Princeton University's WordNet and the US Geological Survey's GNIS. They allow the Ontology Agent to use terms provided by the name services to suggest query enhancements such as generalization or specialization. For example, WordNet can provide a collection of synonyms for a term, while GNIS translates a physical place in the US into latitude and longitude coordinates that are required by a data source such as TerraServer. Other appropriate name and translation services can be added in a modular fashion, and the domain model would be updated to accommodate new concepts and relationships. We now discuss the various sources used by the Ontology Agent.

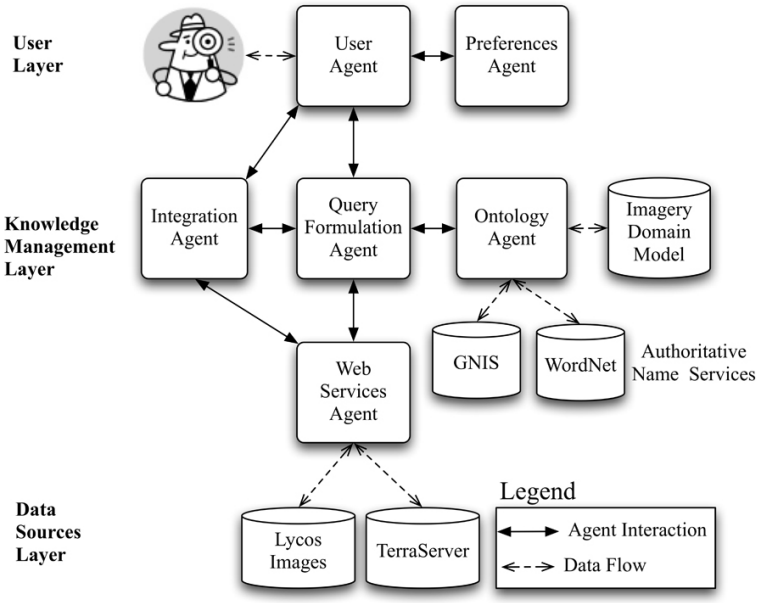

Fig. 1. The Knowledge Sifter Agent-Based Web Services Architecture

Imagery Domain Model Schema. The OWL schema for the imagery domain model, or ontology, is depicted as a UML diagram in Figure 2. The class Image is defined as having *source*, *content*, and file descriptive *features*. Subcategories of content are *person*, *thing*, and *place*. Since we are primarily interested in satellite and geographic images, the class *place* has two general attributes, *name* and *theme*, together with the subclasses *region* and *address*. The Region is meant to uniquely identify the portion of the Earth's surface where the place is located, either by a *rectangle* or a *circle*. In the case of a rectangle we need two latitude values (*north* and *south*) and two longitude values (*east* and *west*), while to specify a circle we need the *latitude* and *longitude* of its center point, and a *radius.* The *address* of our location is identified by *country*, *state*, *city*, *zip code* and *street.* Each image belongs to a specific online source, the *server*, and has *URI-1* as a unique identifier, together with a secondary *URI-2* for a thumbnail (if any). Some qualitative and quantitative attributes are also modeled as subclasses of the general class *features*, namely *resolution* (in square meters per pixel), *projection* and *datum* (for future GIS utilizations), a *date* range, and image *size* (with *height* and *width* expressed in pixels).

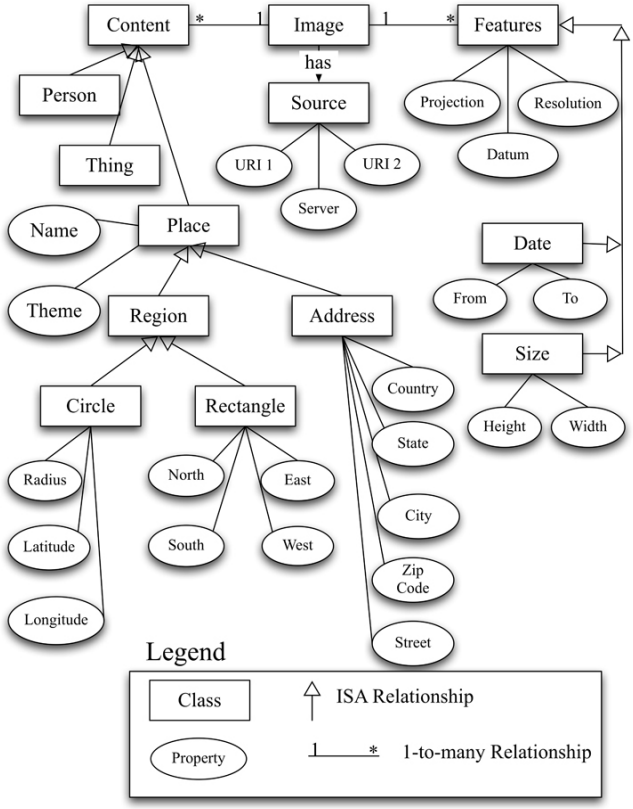

Fig. 2. Ontology Schema in Unified Modeling Language

OWL Schema Specification. An OWL [55] specification of the imagery ontology utilizing RDF and XML was written to provide the conceptual schema of terms and relationships to assist users in instantiating queries. The instantiated schema was passed among the agents and was updated based on decisions and actions taken. To better illustrate how the taxonomy tree in Figure 1 can be converted into an OWL file, here follow some definitions extracted from of the RDF code. The first example shows how to specify that the class *Image* is participating in a relationship between the class *Content*. Such relationship is defined using the OWL nametag "ObjectProperty":

```
<owl:Class rdf:ID="Image">
    <owl:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="Content">
    <owl:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
```

```
</owl:Class>
  <owl:ObjectProperty rdf:ID="HasContent">
       <owl:domain rdf:resource="#Image"/>
       <owl:range rdf:resource="#Content"/>
   </owl:ObjectProperty>
```

As a further example, it is shown how to define a subclass of an OWL class. In this case *Person* and *Thing* are set to be subclasses of the class *Content:*

```
<owl:Class rdf:ID="Person">
       <owl:subClassOf rdf:resource="#Content"/>
</owl:Class>
<owl:Class rdf:ID="Thing">
       <owl:subClassOf rdf:resource="#Content"/>
</owl:Class>
```

Each class or attribute can have a specific Datatype, which can be a simple XMLS type as in the following example, or a customized type created with the appropriate XMLS tags. The following specifies that the class *Person* has type "string":

```
<owl:DatatypeProperty rdf:ID="PersonName">
  <owl:domain rdf:resource="#Person"/>
  <owl:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

The full OWL schema specification may be accessed at: http://www.scs.gmu.edu/~adamiano/csi710/ontology.txt

OWL schema extensions. The OWL schema depicted in Figure 1 captures the concepts that are needed to represent the data and metadata coming from our ontological sources (see section 3.4) and data sources (see section 3.6). However, it would be possible to manipulate and add other portions of ontologies coming from different sources and to integrate them with the existing OWL file. Using the power of RDF, it is reasonable to think of a "library" of ontologies, linked together by RDF relationships between the main upper-level classes. This library could be represented with a high-level RDF document in which the links between the ontologies were hard coded. On the other hand, that would require the Ontology Agent to support some intelligent search among different OWL files. This feature has not been included in the current Knowledge Sifter prototype, but is planned for future research.

Comments on Geospatial Standards and the Imagery Ontology. To enable the inter-operability and extensibility of our ontology and OWL specification, we evaluated the suitability of several existing and developmental metadata standards and image ontologies. Most were either too simplistic or far too detailed for our needs. However, two of the more detailed standards were examined in depth. The Content Standards for Digital Geospatial Metadata (CSDGM), FGDC, June 1994, and the International Standards Organization Technical Committee 211 Metadata Standards (ISO/TC 211). The ISO/TC 211 Metadata Standards were chosen despite being a work in progress. The documentation included a mapping from CSDGM to ISO. In fact, the ISO documentation included a recommendation to continue using the

CSDGM until completion and approval. Direct and indirect participation and contributions are made to ISO/TC 211 by NGA, NIST, FGDC, INCITS, OGIS, and other academic, commercial, and governmental entities from several different countries. Furthermore, federal standards such as Spatial Data Transfer Standard (SDTS), FIPS 173, 1992, and many international standards such as The Digital Geographic Information Exchange Standard (DIGEST), January 1994, and International Hydrographic Organization Transfer Standard for Digital Hydrographic Data (IHO S-57), October 1995, were considered and used in the development.

The three documents of primary interest are currently still in development. ISO 19115 is the Geographic Information/Geomatics Metadata Standard. ISO 19139 is the XML Schema for the Dataset Metadata Implementation Standard for ISO 19115. ISO 19115 Part 2 is an Extension for inclusion of Imagery and Gridded Data. The purpose of these documents is to establish standards for geospatial information concerning "objects or phenomena that are directly or indirectly associated with a location relative to the Earth".

First, we developed a rudimentary ontology, identifying metadata requirements necessary to support access to our two sources. Our intent was to generate an OWL ontology by mapping our metadata requirements to specific elements in the ISO documents. For example, TerraServer has a metadata schema for its own database, and links to the data providers original metadata for data lineage.
For expediency, we implemented our proof-of-concept ontology knowing that we could convert the vocabulary and data types to be fully ISO compliant should those standards be supported by our sources in the future.

3.4 Authoritative Name Services

There are two authoritative name services used by the Ontology Agent: WordNet and GNIS.

Princeton University's WordNet. WordNet is a lexical database for the English language [44]. WordNet is already integrated in the existing version of Web Sifter. When the initial query instance, specifying whether a person, place, or thing, is sent to the Ontology Agent, it then consults WordNet to retrieve synonyms. The synonyms are provided to the Query Formulation Agent to request that the user select one or more synonyms. The decision is communicated to the Ontology Agent which updates the appropriate attribute in the instantiated version of the OWL schema. If the attribute value is the name of a class of type *place* then the Ontology Agent passes the instance to the USGS GNIS.

USGS Geographic Names Information System (GNIS). The USGS GNIS is a database of geographic names within the United States and its territories [54]. GNIS was developed by the USGS and the U.S. Board on Geographic Names to meet major national needs regarding geographic names and their standardization and dissemination. It is an integration of three separate databases, the National Geographic Names Data Base, the USGS Topographic Map Names Data Base, and the Reference Data Base. Records within the database contain feature name, state, county, geographic coordinates, USGS Geographic Map name, and others.

The user's search string is passed to the QueryGNIS web service. An optional *state* can be included to further localize the query. Then a query to the GNIS server is created by generating an appropriately formatted URL that contains the query data. The returned webpage is then parsed for results. There are four possible results pages, no result, a single result, multiple results, and too many results. Each page type is parsed differently. The GNIS limits the return values to less than 2000. When the number of results is greater than that threshold, the number of results is returned but no details are given. The web service user can obtain all these results. Results are then passed to the User Agent to select the preferred location.

For future development, the GNIS the database is available online and could be downloaded and implemented locally. In addition, other authoritative services should be included to yield a broader geographic coverage. For example, the GEOnet Names Server [46] covers much of the world.

3.5 Query Formulation Agent

The user indicates an initial query to the Query Formulation Agent. This agent, in turn, consults the Ontology Agent to refine or generalize the query based on the semantic mediation provided by the available ontology services. Once a query has been specified by means of interactions among the User Agent and the Ontology Agent, the Query Formulation Agent decomposes the query into subqueries targeted for the appropriate data sources. This involves semantic mediation of terminology used in the domain model ontology and name services with those used by the local sources. Also, query translation is needed to retrieve data from the intended heterogeneous sources.

For example, if the user specifies the domain of his search as *place*, Lycos and TerraServer will be chosen. In cases of *person* and *thing*, only Lycos will be chosen. In the case of person and thing, the user is asked to choose a specific meaning from the list retrieved from WordNet, and then the synonym set and hypernym set regarding that particular meaning are retrieved. Synonyms can be chosen as alternate names. Hypernyms can be used to generalize the user's concept. The terms chosen by the user are used to query Lycos. For example, if the users specifies the concept 'Rushmore' the following synonym set is returned by WordNet:

```
Rushmore, Mount Rushmore, Mt. Rushmore -- (a mountain
in the Black Hills of South Dakota; the likenesses of
Washington and Jefferson and Lincoln and Roosevelt are
carved on it)
```

In this case, the synonym set {Rushmore, Mount Rushmore, Mt. Rushmore} and the hypernym set {Mountain Peak} are retrieved from WordNet. If user chooses "Mount Rushmore" and "Mountain Peak", two different queries, "Mount AND Rushmore" and "Mountain AND Peak" are posed to Lycos, because the Lycos image search doesn't support the logical connector "OR" in search terms.

In the case of place, the user-selected synonym set and hypernym set are requested from GNIS server using a similar approach, that is, the queries ("Mount AND Rushmore" and "Mountain AND Peak") are posed to the GNIS server in order to collect a list of locations from which the user can choose. The user can specify a state to restrict the GNIS results. After the user chooses one specific location, the name of

the location is also used to submit queries to the Lycos server. Concurrently, a query is sent to TerraServer Web service with the appropriate latitude and longitude for the selected place.

In our future research, we will endow the Query Formulation Agent with more rules and policies to help it to make more intelligent decisions about query specification and query optimization. For example, in the case of image databases, a strategy might be to query the image metadata, retrieve and view thumbnails, and then request the collection of selected images. In addition, Knowledge Sifter will have a repository of processed queries, instantiated and annotated according to the OWL schema. This information will be used by the Query Formulation Agent as a Case Base that can be searched and the results reused. For example, a user query might be specified in stages, and the Case Base could be used to retrieve a relevant query processing strategy, send a request to the Web Services Agent and the results returned for user consideration. If needed, the Ontology Agent could assist in query enhancement as described above.

3.6 Web Services Agent

The main role of the Web Services Agent is to accept a user query that has been refined by consulting the Ontology Agent, and decomposed by the Query Formulation Agent. The Web Service Agent is responsible for the choreography and dispatch of subqueries to appropriate data sources, taking into consideration such facets as: user preference of sites; site authoritativeness and reputation; service-level agreements; size estimates of subquery responses; and quality-of-service measures of network traffic and dynamic site workload.

The Web Services Agent transforms the subqueries to XML Protocol (SOAP) requests to the respective local databases and open Web sources (TerraServer or Lycos) that have Web Service published interfaces; this is the case for the TerraServer, while Lycos provides an HTTP interface.

3.7 Data Sources (Lycos and TerraServer)

Lycos. The Lycos server supports keyword-based image search via the web page http://multimedia.lycos.com. It makes use of both an image server and external data sources such as web pages for the image search. For a Lycos image search, no advanced search is supported and only conjunctions of terms are used. Therefore, the user cannot specify the image metadata such as *size* or *resolution*, so the results of search are limited. To address these problems the Query Formulation Agent generates a collection of conjunctive and disjunctive queries, while the evaluation and ranking process is left to the Integration Agent.

TerraServer. The TerraServer is a technology demonstration for Microsoft. There is a Web Service API for TerraServer. TerraServer is an online database of digital aerial photographs (DOQs – Digital Orthophoto Quadrangles) and topographic maps (DRGs – Digital Raster Graphics). Both data products are supplied by the U.S. Geological Survey (USGS). The images are supplied as small tiles and these can be made into a

larger image by creating a mosaic of tiles. The demonstrator at terraserver-usa.com uses a mosaic of 2x3 tiles.

Our purpose is to take the ontology-enhanced query and generate specific sub-queries for the TerraServer metadata. The resulting image identifiers and their metadata are wrapped into an instance of our image ontology. And an array of these is returned to the Web Service Agent to compile with other results.

Since our ontology currently specifies a single data point and a single resolution and the TerraServer only returns 200x200 pixel images, each query effectively returns at most one image (DOQ) and one map (DRG). We do want to give the user a bit more than that. So we can return image tiles surrounding our target and in adjacent resolutions.

In the future, there should be support for a range of values. Also, the service should consider generating mosaics of TerraServer image tiles, so the user can get bigger image results than just the 200x200 image. TerraServer is setup to allow this, but the assembly needs to occur on the application side.

3.7 Integration Agent

The Integration Agent is responsible for compiling the sub-query results from the various sources, ranking them according to user preferences, as supplied by the Preferences Agent, for such attributes as: 1) the authoritativeness of a source which is indicated by a weight – a number between 0 and 10 – assigned to that source, or 2) the weight associated with a term comprising a query. The following are the detail specification of similarity value assessment rules for each criterion.

Name:

If the name of the image exactly matches with user's term, then assign the value 10, a perfect match.

Else if the name of the image exactly matches one of synonym sets, then assign the value, 7.

Else if the name of the image includes the user's term, then give the value, 5.

Else if the name of image includes the one of synonym sets, then give the value, (10-5)/(# of Synonyms) for each synonym term.

Theme:

If the image theme (photo or map) matches the user's preference, then assign 10, else assign 0.

Location:

In case of a circle: Value = $\mathrm{Max}(1-\frac{\text{Distance}}{\text{Radius}},0)$

In case of rectangle: Value = $\mathrm{Max}(1-\frac{2*\text{Distance}}{\text{Diagonal}},0)$

Date and Time:

If the time the image was taken is in period of user's preference, then assign 10,

Else assign a value between [0-9] according to the time difference between the user's preference and the date and time the image was taken.

Size:

Similarity Value = 10 * (width_diff * height_diff), where width_diff and height_diff are normalized value according to the size differences.

Resource:

If the source (Lycos or Terra) of image matches with user's preference, then give 10, else give 0.

Total Similarity:

$$\text{Value} = \sum_{c} \text{Sim}(c) * \text{Weight}(c)$$

where c denotes each criterion, and Sim(c) and Weight(c) denote similarity and user preference weight for each criterion, respectively.

The Integration Agent calculates each result's similarity by normalizing the total similarity value, ranks these results according to their similarity values, and returns the ranking information to the User Agent.

4 End-to-End Scenario

Consider the following scenario in which a user wishes to search for the term 'Rushmore'. This scenario is also used in the demonstration of the proof-of-concept prototype.

1. The user provides the User Agent with a query: 'Rushmore'.
2. The user identifies the term is being either a person, place or thing via radio buttons in the query form, see Figure 4. The user has chosen 'Place'.
3. The User Agent passes the query to Query Formulation Agent.
4. The Query Formulation Agent invokes the Ontology Agent to instantiate an OWL schema for the 'Place' with Name = 'Rushmore'.
5. The Ontology Agent chooses a service agent based on the initial query. In this case, it requests from WordNet a list of concepts for 'Rushmore'. WordNet then passes the results back to the Ontology Agent which then passes the results to the User Agent via the Query Formulation Agent for the user decision.
6. The user chooses the 'Mount Rushmore' concept, which has three synonyms ('Rushmore', 'Mt. Rushmore', and 'Mount Rushmore').
7. The Ontology Agent then submits the synonym set to the USGS Geographic Name Information Server and receives a list of candidate geographic coordinates.
8. The list of candidate coordinates is sent to the Query Formulation Agent and the user chooses the desired location.
9. The Ontology Agent then updates the OWL schema instance with the chosen latitude and longitude.
10. The Query Formulation Agent then passes the fully-specified query to the Web Service Agent.
11. The Web Services Agent forwards appropriate sub-queries to both Lycos and TerraServer. The TerraServer and Lycos data sources are queried, and the results are sent back to the Web Services Agent. The results are compiled into new OWL instances that describe image metadata.
12. All results are combined and sent to the Query Formulation Agent.
13. The Query Formulation Agent sends the result sets and the original query to the Integration Agent for ranking.

14. Within the Integration Agent the image metadata for each returned item is ranked using the weights and preferences provided by the Preferences Agent. The Preferences Agent maintains the user preferences.
15. The Integration Agent generates a score for each image result, and returns the scored list to the User Agent.
16. The User Agent then sorts the results by ranking and presents them to the user.
17. The user can then select an item from the list to download and view the image.

5 Knowledge Sifter Proof-of-Concept Implementation

The Knowledge Sifter proof-of-concept prototype has been implemented as an agent-based system, in which each agent is a Web service. Knowledge Sifter has been implemented as a web-application using the ASP and C# languages based on Microsoft .Net Framework. The W3C Document Object Model (DOM) utilized to deal with the XML representation of data by using MSXML technology. Each Agent implemented and deployed as a Web Service through the automation of Microsoft Visual Studio .Net and IIS server using a temporary UDDI. Furthermore, the automation provides the SOAP and WSDL specifications of each agent (Web Services).

Figure 3 shows the User Preferences pane in which users specify preferences and weights associated with semantic names, location, dates, image size, themes, and data sources.

Figure 4 shows the Web page for 1) query formulation, 2) consultation with WordNet and the selection of the synonyms for 'Rushmore" and 3) the results obtained from GNIS for those synonyms.

Figure 5 shows the search results page with image thumbnails, and attributes such as Theme, Location, Data and Time, Size, Resource (data source) and Total Similarity ranking.

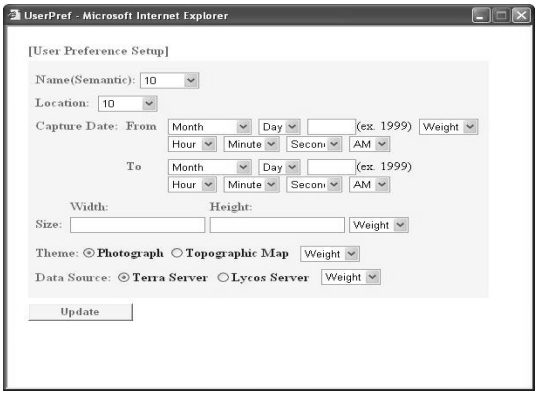

Fig. 3. User Preferences Pane

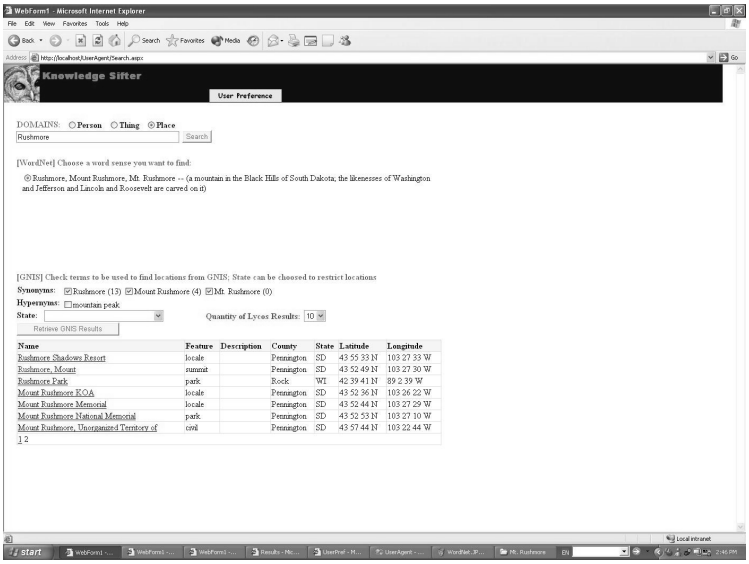

Fig. 4. Knowledge Sifter Query and Ontology Interface

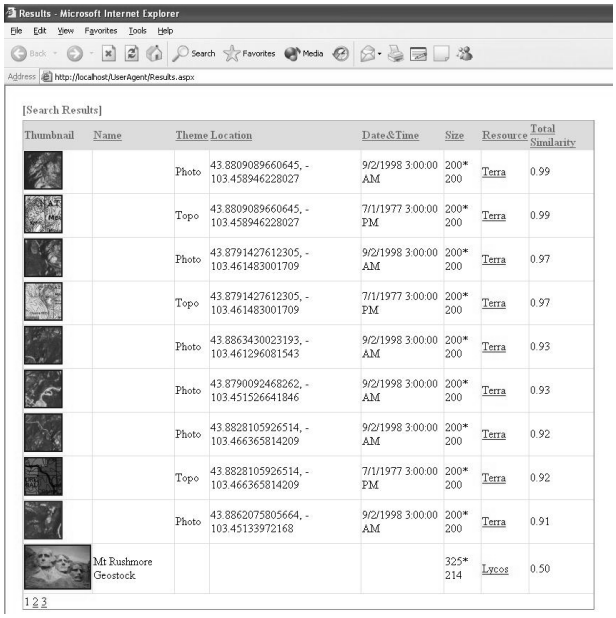

Fig. 5. Knowledge Sifter Search Results Page showing thumbnails of images ranked by similarity in descending order, showing sources, size and theme attributes

6 Conclusions

This paper introduces the Knowledge Sifter agent-based architecture and a proof-of-concept implementation to access heterogeneous data sources.

The concept of a *domain ontology* is central to our approach, in fact, we envision a collection of cooperating ontological sources, accessed by the Ontology Agent that allow a user to pose queries to those data sources with needing to know the location of the supporting data, nor how the ontological concepts are materialized through an integration and ranking process.

The ontology is specified in the Web Ontology Language (OWL) and an XML-instance of the schema is passed among agents in Web services. This permits the agent to annotate the XML-instance so as to provide the *data/knowledge lineage* of a query, including the user preferences, the original query and its enhancements, decisions regarding query decomposition and the choreography of subqueries. This metadata will be stored in a case repository for reuse. We term the concept having the history of an object traveling with it from agent-to-agent, its *digital-DNA*. We intend to expand this notion if future research. Basically, the idea is that a data-object will have its associated digital-DNA, so that it is equipped with knowledge regarding potential interactions with other objects, protocols for negotiation with agents, and rules that determine its behavior. The knowledge required for digital-DNA would be provided by the domain model schema and rules, as well as additional knowledge based on the type of object. This concept will become more important in the context of Grid databases.

Each agent is itself implemented as a Web service. As the Web evolves to the Semantic Web, we will see more ontologies developed in OWL, and we envision that mini-ontologies will be integrated into the existing ontology by means of a collection of RDF-relationships. The inclusion of a new data source will involve appropriate metadata mappings. Our current research includes the specification of a methodology using XTM Topic Maps [4, 50], to specify and merge multiple complementary ontologies in a "plug-and-play" fashion [45].

Future plans include enhancing the capabilities of the existing agents, as well as the investigation, specification, design, development and testing of new agents that will play the role of "staff" agents. These include a Web Services Choreography Agent, a Quality-of-Service Agent, and an Ontology Curation Agent, to deal with workflow management [39], system performance [41-43], and the evolution of the ontology via learning [38]. We are also investigating methods and tools for the dynamic configuration [23] of new sources into the Knowledge Sifter federation.

Finally, we plan to investigate Repository Services, based on XML-database technology, to store and manage the various artifacts that are produced within Knowledge Sifter, including queries, ontology schema instances, lineage, ranking, etc.

Acknowledgements

This work was sponsored by a NURI from the National Geospatial-Intelligence Agency (NGA). This work was also supported in part by the Advanced Research and

Development Activity (ARDA). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U. S. Government.

References

[1] Ankolenkar, A., Hutch, F. and Sycara, K., Concurrent Semantics for the Web Services Specification Language DAML-S. in *Fifth International Conference on Coordination Models and Languages (Coordination 2002)*, (York, UK, 2002).

[2] Aridor, Y., Carmel, D., Lempel, R., Soffer, A. and Maarek, Y.S., Knowledge Agent on the Web. in *Proceedings of the 4th International Workshop on Cooperative Information Agents IV*, (2000), 15-26.

[3] Berners-Lee, T., Hendler, J. and Lassila, O. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities *Scientific American*, 2001, 34-43.

[4] Biezunski, M. Introduction to the Topic Map Paradigm. in Park, J. and Hunting, S. eds. *XML Topic Maps: Creating and Using Topic Maps for the Web*, Addison Wesley, Boston, 2003.

[5] Cercone, N., Hou, L., Keselj, V., An, A., Naruedomkul, K. and Hu, X. From Computational Intelligence to Web Intelligence *IEEE Computer*, 2002, 72-78.

[6] Chakrabarti, S., Berg, M.v.d. and Dom, B., Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. in *Proceedings of the Eighth International WWW Conference*, (1999), Elsevier, 545-562.

[7] Chinnici, R., Gudgin, M., Moreau, J.-J. and Weerawarana, S. Web Services Description Language (WSDL) Version 1.2 (http://www.w3.org/TR/wsdl12/), W3C, 2002.

[8] Decker, S., Erdmann, M., Fensel, D. and Studer, R., ONTOBROKER: Ontology Based Access to Distributed and Semi-Structured Information. in *Database Semantics: Semantic Issues in Multimedia Systems (Proceedings of the 8th Working Conference on Database Semantics)*, (New Zealand, 1999).

[9] Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M. and Horrocks, I. The Semantic Web: the roles of XML and RDF. *IEEE Internet Computing*, *4* (5). 63-73.

[10] Denker, G., Hobbs, J., Martin, D., Narayanan, S. and Waldinger, R., Accessing Information and Services on the DAML-Enabled Web. in *Second International Workshop on Semantic Web (SemWeb'2001)*, (Hong Kong, China, 2001).

[11] Fensel, D. Ontology-Based Knowledge Management *IEEE Computer*, 2002, 56-59.

[12] Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H.-P., Staab, S., Studer, R. and Witt, A., On2broker: Semantic-Based Access to Information Sources at the WWW. in *Proceedings of the World Conference on the WWW and Internet (WebNet 99)*, (Honolulu, Hawaii, USA, 1999), 25-30.

[13] Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H.-P., Studer, R. and Witt, A. On2broker: Lessons Learned from Applying AI to the Web, Institute AIFB, 1998.

[14] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L. and Patel-Schneider, P.F. OIL: an ontology infrastructure for the Semantic Web *IEEE Intelligent Systems*, 2001, 38-45.

[15] Finin, T., Fritzson, R., McKay, D. and McEntire, R., KQML as an Agent Communication Language. in *International Conference on Information and Knowledge Management (CIKM-94)*, (1994), ACM Press.

[16] Gruber, T.R. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, *43* (5-6). 907-928.

[17] Gruber, T.R. A translation approach to portable ontologies. *Knowledge Acquisition, 5* (2). 199-220.
[18] Gu, J., Pedersen, T. and Shoshani, A., OLAP++: Powerful and Easy-to-Use Federations of OLAP and Object Databases. in *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases*, (Cairo, Egypt, 2000), Morgan Kaufmann, 599-602.
[19] Heflin, J. and Hendler, J. A portrait of the Semantic Web in action *IEEE Intelligent Systems*, 2001, 54-59.
[20] Hendler, J. Agents and the Semantic Web. *IEEE Intelligent Systems, 16* (2). 30-37.
[21] Hendler, J. DAML: DARPA Agent Markup Language effort, (http://www.daml.org/), 2002.
[22] Hjelm, J. *Creating the semantic Web with RDF : professional developer's guide*. Wiley, New York, 2001.
[23] Howard, R. and Kerschberg, L., A Knowledge-based Framework for Dynamic Semantic Web Services Brokering and Management. in *International Workshop on Web Semantics - WebS 2004*, (Zaragoza, Spain, 2004), IEEE Computer Society.
[24] Huhns, M. Agents as Web Services *IEEE Internet Computing*, July/August 2002.
[25] ICMWG. Intelligence Community Metadata Working Group, 2003.
[26] ISO_TC_211_19115. Geographic Information - Metadata Part 2 - Metadata for imagery and gridded data, 2003.
[27] ISO_TC_211_19139. Geographic Information - Metadata - Implementation Specification, 2003.
[28] Kerschberg, L. Functional Approach to in Internet-Based Applications: Enabling the Semantic Web, E-Business, Web Services and Agent-Based Knowledge Management. in Gray, P.M.D., Kerschberg, L., King, P. and Poulovassilis, A. eds. *The Functional Approach to Data Management*, Springer, Heidelberg, 2003, 369-392.
[29] Kerschberg, L. (ed.), *Knowledge Management in Heterogeneous Data Warehouse Environments*. Springer, Munich, Germany, 2001.
[30] Kerschberg, L. The Role of Intelligent Agents in Advanced Information Systems. in Small, C., Douglas, P., Johnson, R., King, P. and Martin, N. eds. *Advanced in Databases*, Springer-Verlag, London, 1997, 1-22.
[31] Kerschberg, L., Gomaa, H., Menascé, D.A. and Yoon, J.P., Data and Information Architectures for Large-Scale Distributed Data Intensive Information Systems. in *Proc. of the Eighth IEEE International Conference on Scientific and Statistical Database Management*, (Stockholm, Sweden, 1996), IEEE Computer Society Press.
[32] Kerschberg, L., Kim, W. and Scime, A., Intelligent Web Search via Personalizable Meta-Search Agents. in *International Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2002)*, (Irvine, CA, 2002).
[33] Kerschberg, L., Kim, W. and Scime, A. A Semantic Taxonomy-Based Personalizable Meta-Search Agent. in Truszkowski, W. ed. *Workshop on Radical Agent Concepts*, Springer-Verlag, Tysons Corner, 2002.
[34] Kim, W., Kerschberg, L. and Scime, A. Learning for Automatic Personalization in a Semantic Taxonomy-Based Meta-Search Agent. *Electronic Commerce Research and Applications (ECRA), 1* (2).
[35] Kim, W., Kerschberg, L. and Scime, A., Personalization in a Semantic Taxonomy-Based Meta-Search Agent. in *International Conference on Electronic Commerce 2001 (ICEC 2001)*, (Vienna, Austria, 2001), Elsevier Science.
[36] Klien, M. XML, RDF, and relatives *IEEE Intelligent Systems*, 2001, 26-28.
[37] Lassila, O. and Swick, R. Resource Description Framework (RDF) model and syntax specification (http://www.w3.org/RDF), World Wide Web Consortium, 1998.
[38] Maedche, A. and Staab, S. Ontology learning for the Semantic Web *IEEE Intelligent Systems*, 2001, 72-79.

[39] Marinescu, D.C. *Internet-based workflow management: toward a semantic web*. Wiley-Interscience, New York, 2002.

[40] McIlraith, S.A., Son, T.C. and Zeng, H. Semantic Web Services *IEEE Intelligent Systems*, 2001, 46-53.

[41] Menascé, D.A. QoS Issues in Web Services *IEEE Internet Computing*, 2002.

[42] Menascé, D.A., Almeida, V.A., Riedi, R., Ribeiro, F., Fonseca, R. and W. Meira Jr A Hierarchical and Multiscale Approach to Analyze E-Business Workloads. *Performace Evaluation*.

[43] Menascé, D.A., Dodge, R. and Barbará, D., Preserving QoS of E-commerce Sites Through Self-Tuning: A Performance Model Approach. in *ACM Conference on E-commerce*, (Tampa, FL, 2001).

[44] Miller, G.A. WordNet a Lexical Database for English. *Communications of the ACM*, *38* (11). 39-41.

[45] Morikawa, R. and Kerschberg, L., MAKO: Multi-Ontology Analytical Knowledge Organization based on Topic Maps. in *Fifth International Workshop on Theory and Applications of Knowledge Management*, (Zaragoza, Spain, 2004), IEEE Computer Society.

[46] NIMA. GEOnet Names Server (GNS), http://gnswww.nima.mil/geonames/GNS/index.jsp.

[47] OASIS. Universal Description, Discovery and Integration (http://www.uddi.org/specification.html), OASIS, 2002.

[48] Paolucci, M. and Sycara, K. Autonomous Semantic Web Services *IEEE Internet Computing*, Sept - Oct 2003, 34-41.

[49] Pedersen, T.B., Shoshani, A., Gu, J. and Jensen, C.S., Extending OLAP Querying to External Object Databases. in *CKIM 2000, Proceedings of the 2000 ACM CIKM International Conference on Information and Knowledge Management*, (McLean, VA, 2000), ACM, 405-413.

[50] Pepper, S. and Moore, G. XML Topic Maps (XTM) 1.0, http://www.topicmaps.org/xtm/1.0/, TopicMaps.org, 2001.

[51] Pouchard, L., Cinquini, L., Drach, B., Middleton, D., Bernholdt, D.E., Chanchio, K., Foster, I.T., Nefedova, V., Brown, D., Fox, P., Garcia, J., Strand, G., Williams, D., Chervanek, A.L., Kesselman, C., Shoshani, A. and Sim, A., An Ontology for Scientific Information in a Grid Environment: the Earth System Grid. in *CCGRID 2003*, (2003), 626-632.

[52] Seligman, L. and Kerschberg, L. Federated Knowledge and Database Systems: A New Architecture for Integrating of AI and Database Systems. in Delcambre, L. and Petry, F. eds. *Advances in Databases and Artificial Intelligence, Vol. 1: The Landscape of Intelligence in Database and Information Systems*, JAI Press, 1995.

[53] Sheth, A. and Larson, J. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, *22* (3). 183-236.

[54] USGS. USGS Geographic Names Information System (GNIS), http://geonames.usgs.gov/.

[55]W3C. OWL Web Ontology Language Overview, http://www.w3.org/TR/owl-features/. McGuinness, D.L. and van Harmelen, F. eds., W3C, 2003.

[56] W3C. XML Protocol Working Group (http://www.w3.org/2000/xp/Group/), World Wide Web Consortium, 2002.