Citation for published version:
Patel, M & Duke, M 2004, Knowledge discovery in an agents environment. in CJ Bussler, J Davies, D Fensel & R Studer (eds), The Semantic Web: Research and Applications: First European Semantic Web Symposium, ESWS 2004 Heraklion, Crete, Greece, May 10-12, 2004. Proceedings. Lecture Notes in Computer Science, vol. 3053, Springer, Berlin, Germany, pp. 121-136, The Semantic Web: research and applications, first European Semantic Web Symposium, ESWS 2004, Heraklion, Crete, Greece, 10/05/04. https://doi.org/10.1007/978-3-540-25956-5_9

DOI:
10.1007/978-3-540-25956-5_9

Publication date:
2004

Document Version
Peer reviewed version

Knowledge Discovery in an Agent Environment

Manjula Patel and Monica Duke
{m.patel, m.duke}@ukoln.ac.uk
UKOLN, The Library,
University of Bath, Claverton Down,
Bath BA2 7AY, BANES, UK
Tel: 01225 386547
Fax: 01225 386838

Knowledge Discovery in an Agent Environment

Manjula Patel and Monica Duke
UKOLN
University of Bath, UK
{m.patel, m.duke}@ukoln.ac.uk

Abstract

We describe work undertaken to investigate automated querying of simple forms of ontology by software agents to acquire the semantics of metadata terms. Individual terms as well as whole vocabularies can be investigated by agents through a software interface and by humans through an interactive web-based interface. The server supports discovery, sharing and re-use of vocabularies and specific terms, facilitating machine interpretation of semantics and convergence of ontology in specific domains. Exposure, and hence alignment through ontological engineering should lead to an improvement in interoperability of systems in particular sectors such as education and cultural heritage.

Keywords

Web semantics discovery, ontology server, semantics in agent environments, metadata vocabulary repository, application profiles

1. Background and Motivation

Work in the area of metadata vocabulary repositories has been on-going at UKOLN[42] for some time. As part of the DESIRE[20] and SCHEMAS[36] projects we have been involved in building metadata vocabulary or ontology registries. The primary function of these registries was to provide a publication environment for the disclosure of customised metadata vocabularies also known as *application profiles* [21]. While the DESIRE project had concentrated on a human interface for interrogating the registry, the SCHEMAS project adopted a machine-processible approach based on RDF Schemas (RDFS)[7]. This work was adopted and taken further in the Metadata for Education (MEG) Registry project[31], which was primarily concerned with the UK Education domain and development of the SCART tool[22].

Both the SCHEMAS and MEG registries aimed to provide an environment in which individual terms as well as whole vocabularies can be investigated for adaptations, local usages and relationships with other vocabularies. At present, standard or canonical vocabularies such as the Dublin Core[11] are readily accessible, but this is not the case for application profiles, which are a type of metadata vocabulary that draw on canonical vocabularies and customise them for local use. One major reason for a web-based service is to facilitate the harmonisation of vocabularies or ontology within specific domains such as education, cultural heritage, publishing or rights management and thereby enhance the opportunity for interoperability of systems within such domains. Another use for an ontology server is the automated querying of metadata vocabularies by agents for acquiring the semantics associated with metadata terms. This in turn facilitates the type of reasoning and inference required to fulfil automated service provision and composition in a dynamic environment, and thus help towards the goal of realising the Semantic Web[6,23].

We begin with a discussion that relates ontology and metadata vocabularies and goes on to consider the similarities between ontology servers and metadata vocabulary repositories. We then go on to describe a metamodel for metadata vocabularies which is used as the internal model in our ontology server. The rest of the paper is taken up with describing the implementation, deployment and querying of the ontology server in the context of the Agentcities.NET network environment[1].

2. Ontology and Metadata Vocabularies

From an AI point of view, an ontology serves to define the concepts, terms and relationships used to describe and represent an area or domain of knowledge[16,17]. Ontology aim to capture domain knowledge in a generic way; with differing levels of formality they provide a commonly agreed understanding of a domain, which may be reused and shared across applications and groups. They require a certain degree of ontological commitment for knowledge to be shared. There are several other definitions and typologies of ontology. Some definitions may follow from the way that ontology are built and used; distinctions are made between lightweight and heavyweight ontology, where taxonomies are considered to be one of the former, whereas the latter kind of ontology would include axioms or business rules. For example, Sowa[38] defines a terminological ontology as "an ontology whose categories need not be fully specified by axioms and definition". WordNet [37] is an example of such an ontology. Other distinctions are based on the kind of languages used to implement ontology, such that some ontology are rigorously formal if they are defined in a language with formal semantics, theories and proofs (e.g. of soundness and completeness). Others are highly informal being expressed only in natural language. Some ontology, such as the Dublin Core, are interdisciplinary and are intended to be reusable across domains whilst the majority, such as the IEEE Learning Object Metadata[25] for the education domain, tend to be specific to one area.

Knowledge in ontology is mainly formalized using five kinds of components: classes, relations, functions, axioms and instances. A more detailed description of these components is provided in the OntoWeb Technical Roadmap[32]. However, in this paper we are concerned with only a specific type of simple ontology, referred to as a metadata vocabulary or element set[36]. In the world of digital libraries, a metadata vocabulary or schema declares a set of concepts or terms and their associated definitions and relationships. The terms are often known as elements, attributes and qualifiers. The definitions of the terms provide the semantics which are ideally both human and machine readable. In effect a metadata vocabulary is a manifestation of an ontology, albeit a lightweight one. Such ontology comprise classes, relations and instances. One particular type of vocabulary that we are working with is the *application profile*[21,3]. This is a vocabulary which has been created or tailored for a specific use or application.

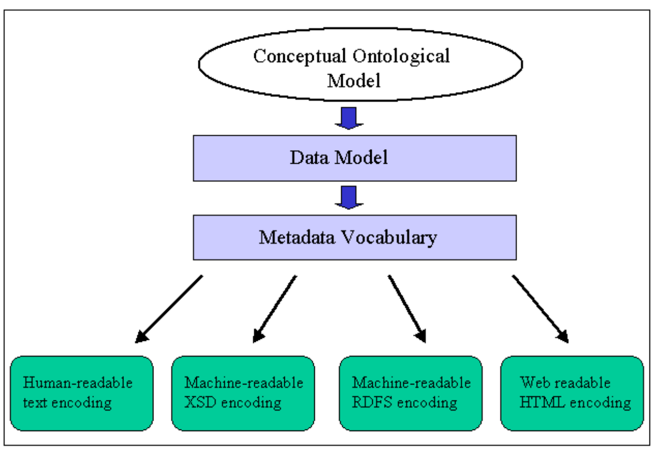

Figure 1: Relationship between ontology, data model, metadata vocabulary and bindings

In particular, an application profile has the following characteristics, it may:

- mix-and-match terms from multiple vocabularies
- specify dependencies (e.g. mandate schemes)
- adapt existing definitions for local purposes
- declare rules for content (e.g. usage guidelines)
- specify whether an element is mandatory, optional or repeatable

Hence we distinguish between a type of vocabulary, or element set, which *declares* metadata terms and a vocabulary, or application profile, which makes *use* of terms declared elsewhere.

Figure 1 shows the relationship between an abstract ontological model, an underlying data model and the metadata vocabulary that may be derived from the two. The metadata vocabulary can in turn be instantiated using differing encoding technologies or bindings.

2.1 Ontology Specification Languages

Semanticweb.org[30] provides a historical perspective on the representation of ontology on the Web. The most influential over the past few years has been, the Resource Description Framework (RDF)[28]. This is a general-purpose modeling framework for representing information on the Web. It consists of graph structures made up of object, predicate and subject triples. The RDF Schema specification (RDFS) [7] describes how to use RDF in order to describe RDF vocabularies. A major benefit of RDFS is that it provides for the expression of relationships between semantic terms. It is notable that RDF seems to be emerging as the lowest common denominator and a basis for many other ontology description languages.

More recently, the Web Ontology Language (OWL)[33] is being designed by the W3C Web Ontology Working Group[45] in order to provide a language which can be used by applications that need to process the content of information as opposed to just the syntax. OWL is a semantic markup language for publishing and sharing ontology on the Web. It provides greater machine processibility of web content by supplementing RDFS with additional vocabulary terms for more accurate specification of relationships over and above that of subsumption. The OWL language is a revision of the DAML+OIL web ontology language, it incorporates experience from the design and application uses of DAML+OIL[10]. Different subsets of the OWL language are defined to suit different uses (OWL Lite, OWL DL and OWL Full). OWL has been designed for maximal compatibility with RDF and RDFS, an OWL ontology being represented as a set of RDF triples.

Whilst ontology description languages were in a state of flux, we chose to base our encodings on RDFS which is being used as the basis for more sophisticated specification languages such as OWL.

3. Ontology Servers and Metadata Vocabulary Repositories

A repository of metadata vocabularies enables individual terms as well as whole vocabularies to be investigated for adaptations, local usages and relationships with other vocabularies. Such data mining facilitates analysis of patterns of usage as well as the creation and inference of new information. Furthermore, these types of services are an essential part of ontological engineering procedures if there is to be any hope of convergence of domain level knowledge representation.

Our server has an architecture which can harvest distributed metadata vocabularies from their maintainers over the Web. In this manner authoritative control over particular vocabularies is devolved to their original developers and maintainers. In response to queries, the server responds by providing term-level definitions and usage information, along with contextual annotations. The repository, in effect functions as an indexing engine for dynamically updating, , and serving up the semantics of metadata terms. The context for such a server is the notion of a Semantic Web where any person or organisation can declare a metadata vocabulary and assert a relationship between that vocabulary and any other vocabulary or term on the Web.

Other initiatives within the areas of ontology, ontology representation, storage and exchange have also undertaken reviews of repositories of ontology. The OntoWeb Technical RoadMap[32] lists some of the better known ones. The ontology repositories that are described include those in which ontology are implemented in DAML, Ontolingua[13] and SHOE[19]. More recently, the SWAD Europe Project reviewed RDF storage systems[5] including ones that may include schema and ontological data such as RDFS and DAML+OIL. The DAML Repository[9] is a web-accessible catalogue of ontology expressed in DAML. In addition, work in the area of developing knowledge bases of semantic terms is well established in the AI sector. A notable example is that of Lenat's Cyc[29].

One area in which we have been actively promoting harmonization is that of education in the UK. The Metadata for Education Group (MEG) was formed following a meeting of key UK stakeholders and serves as an open forum for debating the description and provision of educational resources at all educational levels across the UK. This group seeks to reach consensus on appropriate means by which to describe discrete learning objects in a manner suitable for implementation in a range of educational arenas. Preceding work undertaken in the DESIRE and SCHEMAS projects provided the basis for the MEG Registry Project, which adopted a slightly modified data model as described below. The aim of the MEG registry is to provide implementers of educational systems with a means to share information about their metadata vocabularies and to re-use existing schemas. The benefit being a saving in time and effort which would be spent in researching existing vocabularies and in re-inventing terms.

The Meg Registry is implemented as a server based on the RDF toolkit, Redland[4]. Information about the entities described in section 4 and their relationships, is stored and made available in machine-processible format as RDFS. The registry API has been developed in Perl and supports functions such as querying through an HTTP interface. In section 4 we provide an overview of the data model and definitions employed in the MEG Registry project since they have provided the framework for the work described in this paper.

4. A Meta-Model for Metadata Vocabularies

Our ontology server is based on the following model of metadata vocabularies and application profiles, addition details can be found in [22]:

Element Sets are owned and maintained by **Agencies**.
Element Sets are made up of **Elements**.
An **Element Usage** may:

- introduce constraints on the value of an **Element** by associating it with one or more **Encoding Schemes**
- introduce constraints on the *obligation* to use an **Element** (e.g. make its use mandatory) or the *occurrence* of an **Element** (e.g. whether it is repeatable)
- *refine* the semantic definition of an **Element** to make it narrower or more specific to the application domain.

Encoding Schemes constrain the value space of **Elements**.
An **Application Profile** defines a set of **Element Usages** of **Elements** drawn from one or more **Element Sets**.

The server holds information on each of the entities and their relationships:

- **Element Sets** (i.e. on the Element Sets as units, rather than on their constituent Elements), including information on their intended scope/area of use and their relationship to other Element Sets
- the **Elements** which make up those Element Sets, including information on the semantics of the Elements and their recommended usage, and any semantic relationships to other Elements in this or other vocabularies (e.g. the relationship described by the DC[11] concept of "element refinement" or by RDF Schema as a "sub-property" relation)
- **Application Profiles**, including information on their intended scope/area of use and their relationship to other Element Sets
- the **Usages of Elements** which make up Application Profiles, including the Element used, any prescription of Encoding Schemes, and other constraints on element use
- **Encoding Schemes**, which constrain the value space of Elements, including information on their intended scope/area of use; where an Encoding Scheme takes the form of an enumerated list, the **values** prescribed by that Encoding Scheme may be recorded
- the **Agencies** who own/create/maintain Element Sets, Application Profiles, and Encoding Schemes

5. Implementation of an Ontology Server

We have extended the work done in the MEG Registry project to re-deploy the interfaces to the registry within an agent environment, namely the Agentcities.NET network. The existing registry software stores information relating to metadata vocabularies and provides an interface for interacting with the repository of information. We have thus transitioned from a human-centric to an agent-centric environment

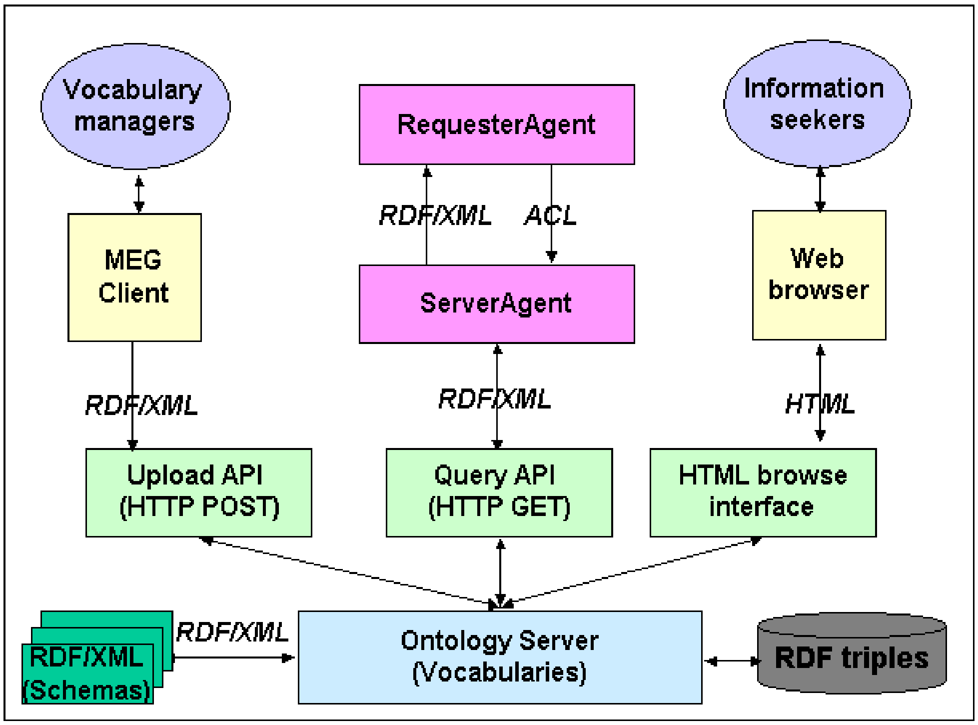

Figure 2: Ontology Server Architecture

Figure 2 is an adapted diagram of the architecture of the MEG Registry. It illustrates the architecture of the ontology server and its deployment in an agents environment. We discuss in more detail the processes, interfaces and protocols involved.

5.1 Ontology Acquisition, Server Population

The contents of the ontology server comprise metadata vocabularies also known as schemas or metadata element sets. The notion of application profiles has been used as the underlying data model for encoding the metadata vocabularies. The vocabularies are modelled within the server as outlined in section 4. The server uses a harvesting paradigm to “pull in” vocabularies which have been encoded using RDF Schemas with XML as the serialisation syntax. Population of the server is achieved by specifying a URI from which an RDFS encoding of a vocabulary is retrieved over HTTP and stored in a database. The advantage of using a harvesting paradigm is that maintenance and authority over an ontology is decentralised and devolved to those committed to its development. The server can be regularly updated in a similar manner to current search engines which crawl the Web in order to index Web pages.

5.2 Deployment of Service within the agentcities.NET Network

Our implementation work has been carried out using the Java Agent Development Environment (JADE)[26]. JADE is one of the recommended platforms for developing agent systems in the Agentcities.NET network. It is a software development platform aimed at creating multi-agent systems and applications conforming to FIPA[14] standards for intelligent agents. It includes two main products, a FIPA-compliant agent platform and a package to develop Java agents. We have developed a *ServerAgent* which runs on the UKOLN agent platform and accepts requests from other agents. It responds to requests by retrieving information from the server, communicating with the server through the server API (over HTTP), and returning the results. Exploration of vocabularies is organised around the entities described by the data model in section 4, these comprise: *agency; element; element set; application profile; encoding scheme* and *element usage*. Results are returned as RDF-encoded data. This is possible since the native store of the server records the element set descriptions as RDF triples.

5.3 Interactive Web Interface

For ease of accessibility, the knowledge base of semantic terms may be explored through an interactive web interface which caters for search, browse and navigation functions[44]. Metadata schema implementers have indicated that a service which allows for searching, browsing and navigation of vocabularies would play an essential role in developing new element sets, as well as serving in the harmonization of vocabularies for specific domains. The screen shots in Figures 3 and 4 illustrate navigation using a web browser. Browsing a category (e.g. elements sets) reveals a list of all the resources of that class, with links to further details.

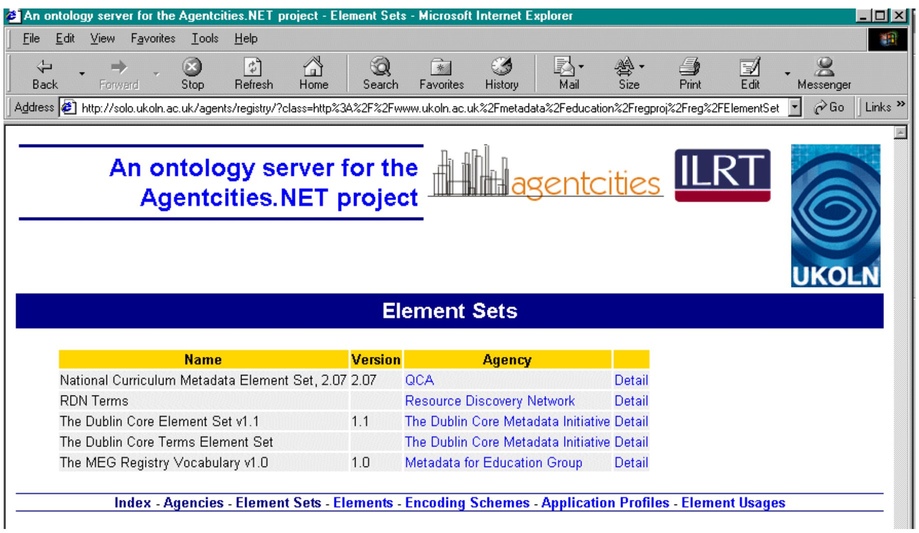

Figure 3: Browsing the list of all element sets in the metadata vocabulary repository

When browsing a specific resource, the details from the RDF description of that resource are displayed, as well as links to related resources.

5.4 Machine Interface

We have developed a software interface to the server to allow agents to query and navigate metadata vocabularies. Below we describe the *ServerAgent* and two examples of requester agents. The *ServerAgent* can carry out search and browse requests on behalf of other agents, and passes on the results from the server to the requester agents.

Searches are carried out within a specific category, such as agency or elements, and the search term is matched with any part of the text between the RDF tags making up a description. If a part of the description matches, the whole description for that resource is returned in the result set. When the description is that of an element, the description of the associated element set is also presented.

Using the browse function, either a whole category is explored, or a named resource from a category is specified. The RDF descriptions for all the resources in a category, or for a single resource are returned respectively.

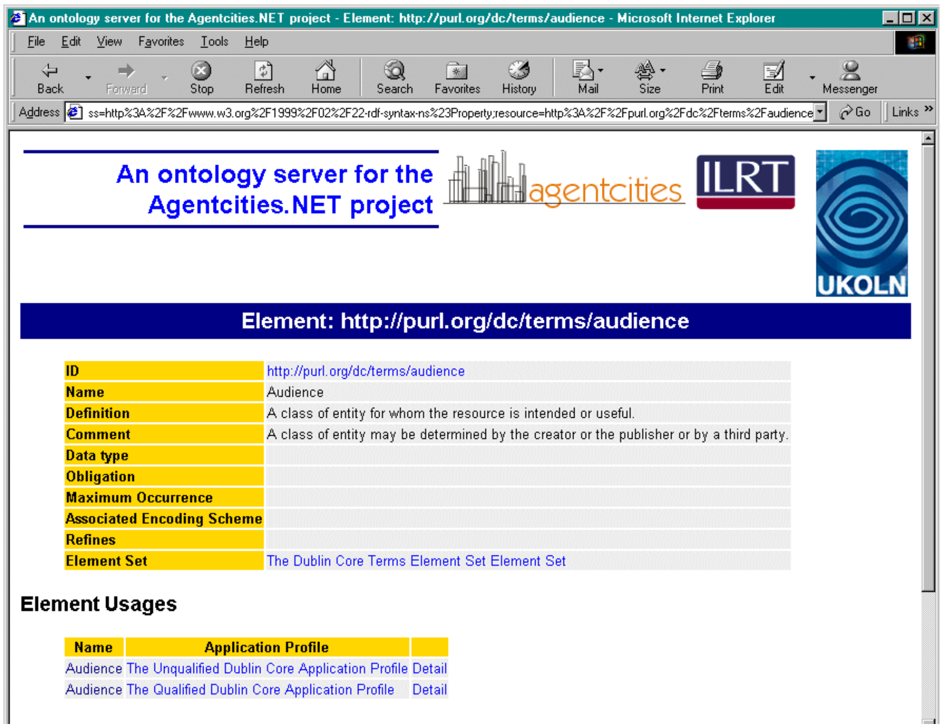

Figure 4: Browsing the details of a specific vocabulary term or element

JADE conceptualizes an agent as an independent and autonomous process that has an identity, possibly persistent, and that requires communication (e.g. collaboration or competition) with other agents in order to fulfill its tasks. This communication is implemented through asynchronous message passing and by using an Agent Communication Language with a well-defined and commonly agreed semantics. JADE queries to the *ServerAgent* are made using the FIPA Agent Communication Language[14]. On receiving a request message, the *ServerAgent*:

1. extracts components of the request (using an ontology)
2. constructs a URL from the request
3. connects to the server using the URL
4. reads the response from the server
5. places the response into a reply message

The *ServerAgent* uses a simple behaviour model to deal with one request at a time, sending a reply before attending to the next request message in the queue. A more complex model of behaviour, for example starting a new agent or behaviour to deal with each request, was unnecessary at this stage, given the simple functionality of the server and the agent. In a service level server, the issue of how to deal with a large number of requests in a timely manner would become important. The performance of a large server capable of complex querying would also have to be taken into account.

5.5 Internal Server Ontology

We have defined a simple ontology (*ServerSearchOntology*) in which requests to the *ServerAgent* can be expressed. This ontology is intended to encapsulate the simple kinds of requests supported by the server and is not intended to be an exhaustive or comprehensive ontology for all the kinds of queries that metadata vocabulary repositories should or could support.

The ontology consists of two *Action* concepts, *ReturnSearchResults* and *ReturnBrowseResults*. The *ReturnSearchResults* action emulates a search request through a web browser; *ReturnSearchResults* has a *searchRequest*, made up of a *Scope* and a *searchTerm*. The scope limits the search for the *searchTerm* to one of the categories specified in section 4. *ReturnBrowseResults* emulates the browsing action carried out through the web browser. Thus a *browseRequest* takes a *Scope* and a specific resource URI. The resource URI identifies a specific instance of the entity (e.g. a particular agency) and if a resource URI is provided in the browse request, the RDF description for that resource alone is returned. If no resource URI is specified, the RDF descriptions of all the instances of that category are returned in a list. The examples below illustrate this behaviour.

Example 1: A **search** for the **term** "audience" in the **element** category

```
(        (action
                  (agent-identifier :name UKOLNServer@solo.ukoln.ac.uk:1099/JADE)
                  (ReturnSearchResults
                           (Search : Scope element : SearchTerm audience)
         ) )
)
```

```
<rdf:Description rdf:about="http://purl.org/dc/terms/mediator">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label>Mediator</rdfs:label>
    <rdfs:comment>A class of entity that mediates access to the resource and for
whom the resource is intended or useful.</rdfs:comment>
    <reg:useComment>The audience for a resource in the education/training domain
are of two basic classes: (1) an ultimate beneficiary of the resource (usually a
student or trainee), and (2) frequently, an entity that mediates access to the
resource (usually a teacher or trainer).  The mediator element refinement
represents the second of these two classes.</reg:useComment>
    <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/terms/audience"/>
    <reg:isElementOf
rdf:resource="http://www.ukoln.ac.uk/metadata/education/regproj/reg/elementSet/dcte
rms"/>
  </rdf:Description>

<rdf:Description rdf:about="http://purl.org/dc/terms/audience">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label>Audience</rdfs:label>
    <rdfs:comment>A class of entity for whom the resource is intended or
useful.</rdfs:comment>
    <reg:useComment>A class of entity may be determined by the creator or the
publisher or by a third party.</reg:useComment>
    <reg:isElementOf
rdf:resource="http://www.ukoln.ac.uk/metadata/education/regproj/reg/elementSet/dcte
rms"/>
  </rdf:Description>

<rdf:Description
rdf:about="http://www.ukoln.ac.uk/metadata/education/regproj/reg/elementSet/dcterms
">
    <rdf:type
rdf:resource="http://www.ukoln.ac.uk/metadata/education/regproj/reg/ElementSet"/>
    <dc:title>The Dublin Core Terms Element Set</dc:title>
    <dcterms:created>2000-07-11</dcterms:created>
    <reg:status>DCMI recommendation</reg:status>
    <dc:description>
      The Dublin Core metadata vocabulary is a simple vocabulary intended to facilitate
      discovery of resources.
    </dc:description>
  <reg:responsibleAgency
rdf:resource="http://www.ukoln.ac.uk/metadata/education/regproj/reg/agency/dcmi"/>
    <reg:xmlNamespacePrefix>dcterms:</reg:xmlNamespacePrefix>
    <reg:specification rdf:resource="http://dublincore.org/usage/terms/terms-
latest.html"/>
  </rdf:Description>
```

This search finds two elements: the search term 'audience' is found within the useComment tag of the Mediator element, as well as the Audience element in the Dublin Core Metadata Element Set (the search term is highlighted here for emphasis). Both these elements are part of the Dublin Core Terms element set and the description for the element set is returned at the end.

Example 2: A browse request for a specific resource (http://purl.org/dc/terms/MESH/) from the encoding scheme category.

```
(       (action
                (agent-identifier :name UKOLNServer@solo.ukoln.ac.uk:1099/JADE)
                (ReturnBrowseResults
                 (Browse :Scope encodingscheme  :Resource http://purl.org/dc/terms/MESH)
        ) )
)
```

5.6 Interrogating the ServerAgent

We have implemented two examples of *RequesterAgent*, both of which are driven by a human user and make requests to the *ServerAgent*. These two agents use the *ServerSearchOntology* to communicate requests to the *ServerAgent*, and then display the response returned by the server. Results to queries consist simply of RDFS descriptions wrapped up in a standard agent response. Thus the *ServerSearchOntology* is only used to communicate requests, not responses.

5.6.1 The GUI Agent

This agent presents the user with a graphical interface implemented with Java Swing. JADE provides support for building interfaces for agents which need to interact with users, and provides guidelines on how to implement agents that work within the Java GUI concurrency model. The GUI Interface is realised through two classes:

RequesterAgentGui class extends the *Swing JFrame* class, and defines the appearance of the interface

RequesterGuiAgent class extends the *JADE GuiAgent* class, and defines the behaviours that are instantiated in response to user actions at the interface.

The appearance of the interface is shown in Figure 5.

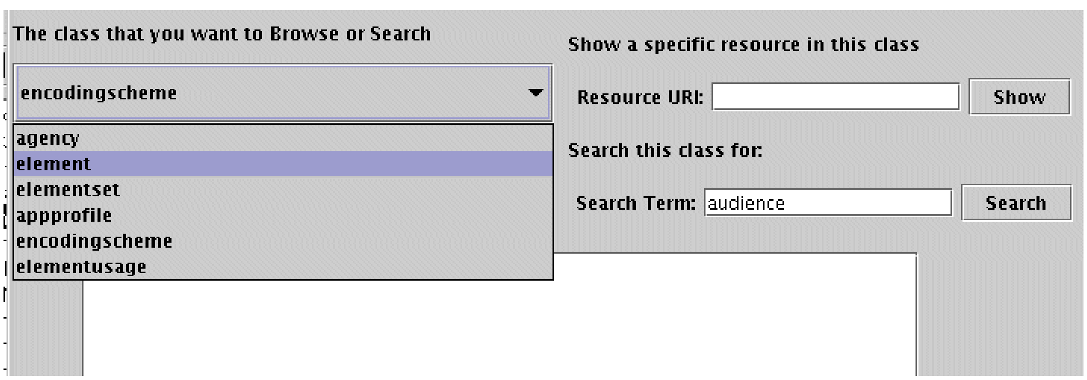

Figure 5: Using the interactive GUI of the *RequesterGuiAgent* to select a category and enter a search term

All browse and search actions through the GUI start by selecting a category. The user can then choose to browse the whole category, or to enter a resource URI for a known resource. Alternatively, the search box can be used to interrogate the server. The three tasks that the interface supports reflect the kinds of requests that can be expressed in the *ServerSearchOntlogy*.

The interface has been designed to support one outstanding request at a time. In theory multiple requests could be launched before the first response arrives, and at present there is no control to prevent this. In practice the system response is sufficiently fast that no major control is required at present to synchronise requests and responses.

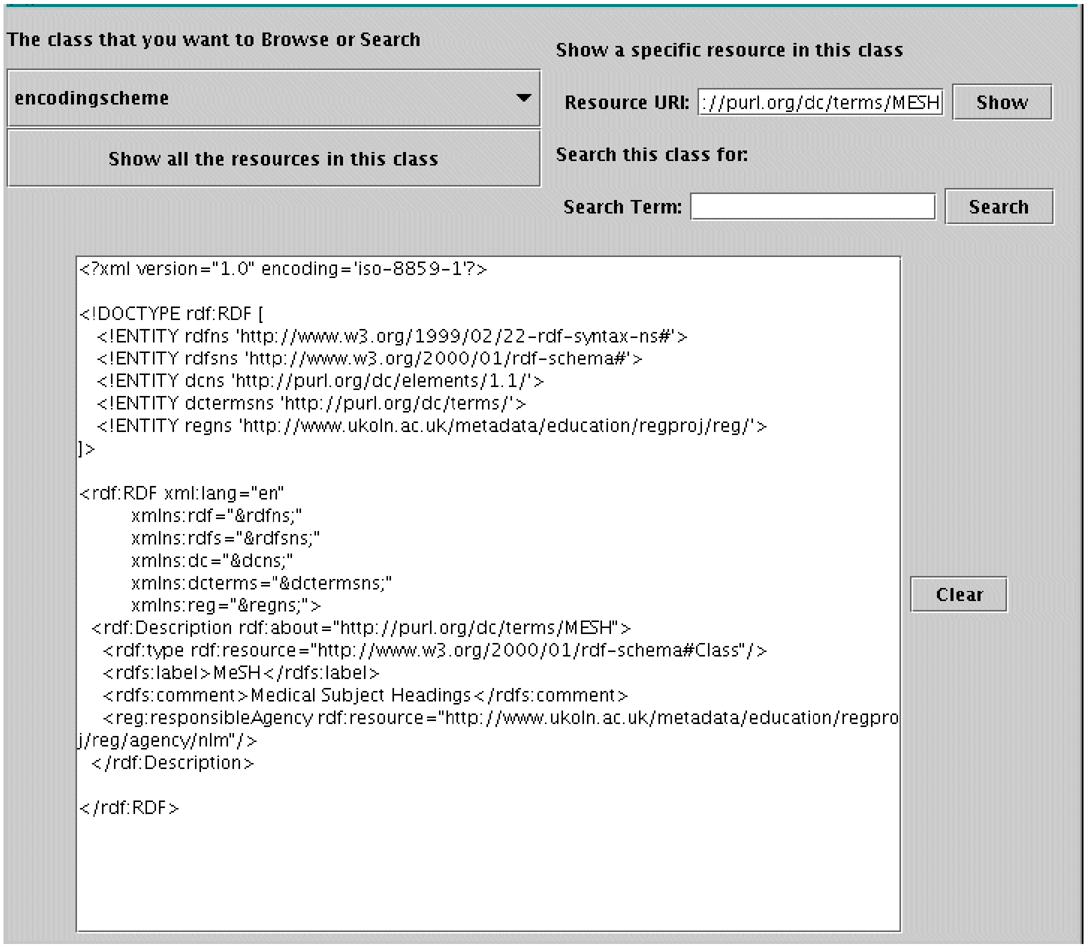

Figure 6. Results are displayed in a window in the GUI.

5.6.2 The Command Line Agent

A second agent class, *ServerRequesterAgent*, has been provided to interact with the user through the command line. On setup this agent first establishes which server the user would like to use, with a choice of either the UKOLN server, or a local one. The agent then prompts the user for input, establishing what kind of transaction the user is performing (browse or search) and its parameters: scope, search term or resource URI:

```
ENTER the local name of the Server agent or press enter to use the UKOLN
Server-->
ENTER s for search or b for browse -->
s
Class to Search ---> element
Enter a SearchTerm ---> audience
```

A suitable request message is then built by the *ServerRequesterAgent* (using the *ServerSearchOntology*) and sent to the *ServerAgent.* The response from the *ServerAgent* is displayed to the user as an RDFS encoding (as in Example 1). The user can then choose to carry out another transaction or terminate the interaction

6. Conclusions and Further Work

We have adapted software for a metadata vocabulary registry to serve as an ontology server which can be queried by agents on the Agentcities.NET network. The contents of the repository comprise metadata vocabularies which may be regarded as simple forms of ontology. It should be noted that the server accepts metadata vocabularies encoded in RDF Schema. Further, the vocabularies need to adhere to the data model described in section 4, which is based on the notion of application profiles. The work presented has advanced the work begun in previous projects to investigate an approach based on automated querying and processing of simple ontology by software agents as well as through human interaction.

Commitment to an ontology exchange format (and vocabulary description language) requires further study. There are various proposals in this area, as reviewed in the OntoWeb Technical Roadmap. Some are the result of various W3C working groups, other proposals may come from the agent community itself. The Agentcities Ontology working group[1] suggests a number of tools that may feed into the study of the adoption of a specific framework, all of which can handle RDF Schemas.

Clearly the *ServerAgent i*s required to interact with other agents as well as the underlying repository. Further research is required to understand precisely which requests from other agents the server should handle. Work is also required to assess models of response. For example, should the *ServerAgent* respond to requests by simply wrapping data retrieved from the server within a content response parameter, or should the response be more structured, possibly entailing additional dialogue between the agents.

As has already been mentioned, there is a need for a service which enables the publication and disclosure of the semantics that are being used in applications and web services. The advantages of providing a machine-processible format are numerous. An ontology server can be queried by software agents roaming the Web in order to retrieve semantics and thereby perform reasoning tasks which would aid automation of the Web and move it on to the next generation of development Ontology servers are an essential part of the infrastructure required to enable the exchange and re-use of vocabularies and individual terms. The provision of semantics in a machine-processible format is a key goal in the road towards semantic interoperability.

Acknowledgements

The work we have described was made possible by a deployment grant awarded to UKOLN under the European Commission funded 5th Framework IST project Agentcities.NET. The software used in this project for the ontology server was originally developed in the MEG Registry project which was funded by JISC and BECTa. The ideas in this project have been developed from work in the DESIRE and SCHEMAS projects. Thanks to Pete Johnston, UKOLN, University of Bath, for help with the MEG Registry software and also the original diagram on which Figure 2 is based. Thanks also to Owen Cliff, Department of Computer Science, University of Bath for assistance with setting up UKOLN's JADE platform.

UKOLN is funded by Resource: The Council for Museums, Archives & Libraries, the Joint Information Systems Committee (JISC) of the Higher and Further Education Funding Councils, as well as by project funding from the JISC and the European Union. UKOLN also receives support from the University of Bath where it is based.

References

[1] Agentcities.NET Website
http://www.agentcities.org/EUNET/, last visited 25th November 2003

[2] M. Annemalai and L. Sterling, Guidelines for Constructing Reusable Domain Ontologies, Ontologies in Agent Systems Workshop, July 2003, Melbourne, Australia
http://oas.otago.ac.nz/OAS2003/papers/oas03-annamalai2.pdf, last visitied 25th November 2003

[3] T. Baker, M. Dekkers, R. Heery, M. Patel, G. Salokhe, What Terms Does Your Metadata Use? Application Profiles as Machine Understandable Narratives, Journal of Digital Information Vol 2 (2), November 2001, http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Baker/baker-final.pdf, last visited 25th November 2003

[4] D. Beckett, The Design and Implementation of the Redland RDF Application Framework, Proceedings of WWW10, Hong Kong, May 2-5 2001, http://www10.org/cdrom/papers/frame.html, last visited 25th November 2003

[5] D. Beckett, SWAD-Europe, Scalability and Storage: Survey of Free Software /Open Source RDF storage systems, http://www.w3.org/2001/sw/Europe/reports/rdf_scalable_storage_report/, last visited 12th November 2003

[6] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, Scientific American, May 2001 http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21, last visited 25th November 2003

[7] D. Brickley and R.V. Guha, RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft October 2003,
http://www.w3.org/TR/rdf-schema/, last visited 10th November 2003

[8] P. Charlton, D. Bonnefoy and N. Huillier, Agents and their Cities, European Conference on Artificial Intelligence 2002, Lyon, France
http://ecai2002.univ-lyon1.fr/show_en.pl?page=en/cfp/ecai_accepted.html, last visited 25th November 2003

[9] DAML Ontology Library
http://www.daml.org/ontologies/, last visited 12th November 2003

[10] DAML+OIL (March 2001) Reference Description, W3C Note 18, December 2001, http://www.w3.org/TR/daml+oil-reference, last visited 12th November 2003

[11] ISO 15386: The Dublin Core Metadata Element Set
http://www.niso.org/international/SC4/sc4docs.html, last visited 12th November 2003

[12] M. Duke and M. Patel, An Ontology Server for the Agentcities.NET Project, October 2003, http://www.agentcities.org/note/00008/, last visited 10th November 2003

[13] A. Farquhar, R. Fikes, J. Rice, The Ontolingua Server: a Tool for Collaborative Ontology Construction. In KAW96, November 1996.
http://www-ksl.stanford.edu/people/axf/papers.html/KSL-TR-96-26.ps, last visited 2nd December 2003

[14] Foundation for Intelligent Physical Agents (FIPA) Homepage
http://www.fipa.org, last visited 27th November

[15] N. Gibbons, S. Harris and N. Shadbolt, Agent based Semantic Web services, Proceedings of the 12th World Wide Web Conference, Budapest, May 2003
http://www2003.org/cdrom/papers/refereed/p455/p455-gibbins.html, last visited 10th November 2003

[16] T.R. Gruber, The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases: in Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference, Cambridge, MA, pp. 601-602, Morgan Kaufmann, 1991.

[17] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing. Presented at the Padua workshop on Formal Ontology, March 1993

[18] R.V. Guha, D. B. Lenat, K. Pittman, D. Pratt, and M. Shepherd., Cyc: A Midterm Report. Communications of the ACM 33, No.8, August 1990.

[19] J. Heflin, J. Hendler, and S. Luke, SHOE: A Knowledge Representation Language for Internet Applications, University of Maryland Technical Report CS-TR-4078 (UMIACS TR-99-71), 1999 http://www.cs.umd.edu/projects/plus/SHOE/pubs/techrpt99.pdf, last visited 2nd December 2003

[20] R. Heery, T. Gardner, M. Day and M. Patel, DESIRE Metadata Registry Framework, 2000, http://www.desire.org/html/research/deliverables/D3.5/, last visited 12th November 2003

[21] R. Heery and M. Patel., Application Profiles: mixing and matching metadata schemas, Ariadne Issue 25, September 2000 http://www.ariadne.ac.uk/issue25/app-profiles/intro.html, last visited 12th November 2003

[22] R. Heery, P. Johnston, D. Beckett and D. Steer, The MEG Registry and SCART: complementary tools for creation, discovery and re-use of metadata schemas. In: Proceedings of the International Conference on Dublin Core and Metadata for e-Communities, October 2002. Florence: Firenze University Press, 2002, pp. 125-132. http://www.bncf.net/dc2002/program/ft/paper14.pdf, last visited 12th Novemver 2003

[23] J. Hendler, Agents on the Semantic Web, IEEE Intelligent Systems, March/April 2001

[24] C. W. Holsapple and K. D. Joshi, A Collaborative Approach to Ontology Design, Communications of the ACM, Feb 2002, Vol 45(2) pp42-47

[25] IEEE Learning Object Metadata (IEEE LOM), http://ltsc.ieee.org/wg12/, last visited 12th November 2003

[26] Java Agent Development Environment (JADE) Homepage http://jade.cselt.it/, last visited 27th November

[27] H. Kim, Predicting How Ontologies for the Semantic Web will Evolve, Communications of the ACM, Feb 2002, Vol 45(2) pp48-54

[28] O. Lassila and R. Swick, Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999 http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, last visited 10th November 2003

[29] D.B Lenat, Cyc: A Large-Scale Investment in Knowledge Infrastructure, Communications of the ACM 38, no. 11 (November 1995).

[30] A. Maedche, B, Motik, L. Stojanovic, R. Studer and R. Volz, An Infrastructure for Searching, Reusing and Evolving Distributed Ontologies, In Proceedings WWW2003, Budapest, Hungary, May 20-24 2003

[31] Metadata for Education (MEG) Registry Project http://www.ukoln.ac.uk/metadata/education/regproj/, last visited 12th November 2003

[32] OntoWeb Technical Roadmap v 1.0 http://babage.dia.fi.upm.es/ontoweb/wp1/OntoRoadMap/index.html, last visited 12th November 2003

[33] D.L. McGuinness and F. van Harmelen, OWL Web Ontology Language Overview, W3C Candidate Recommendation 18 August 2003, http://www.w3.org/TR/2002/WD-owl-features-20020729/, last visited 10th November 2003

[34] M. Patel, Ontology Harmonisation towards the Semantic Web, Poster, in Proceedings of the International Semantic Web Conference, June 2002, Sardinia, Italy

http://www.ukoln.ac.uk/ukoln/staff/m.patel/iswc2002/iswc-poster.pdf, last visited 10th November 2003

[35] SemanticWeb.org, Markup Languages and Ontologies
http://www.semanticweb.org/knowmarkup.html#ontologies, last visited 12th November 2003

[36] SCHEMAS Forum Registry
http://www.schemas-forum.org/registry/, last visited 12th November 2003

[37] D. Soergel, WordNet, Book Review, D-Lib Magazine, October 1998
http://www.dlib.org/dlib/october98/10bookreview.html, last visited 12th November 2003

[38] J.F. Sowa, Building, Sharing and Merging Onotologies: Glossary
http://www.jfsowa.com/ontology/ontoshar.htm#s6, last visited 12th November 2003

[39] S. Staab, R. Struder, H.P Schum and Y. Sure, Knowledge Process and Ontologies, IEEE Intelligent Systems, Jan/Feb 2001, pp26-34

[40] L-K. Soh, Collaborative Understanding of Distributed Ontologies in a Multiagent Framework: Design and Experiments, Ontologies in Agent Systems Workshop, July 2003, Melbourne, Australia
http://oas.otago.ac.nz/OAS2003/papers/oas03-soh.pdf

[41] V. Tamma and T. Bench-Capon, A conceptual model to facilitate knowledge sharing in multi-agent systems, in Proceedings of the Autonomous Agents 2001 Workshop on Ontologies in Agent Systems (OAS 2001). Montreal, May, 29, 2001, pp. 69-76.

[42] UKOLN Metadata Projects
http://www.ukoln.ac.uk/metadata/, last visited 12th November 2003

[43] UKOLN web-page for Agentcities.NET deployment project
http://www.ukoln.ac.uk/metadata/agentcities/, last visited 12th November 2003

[44] UKOLN Ontology Server for the Agentcities.NET network
http://solo.ukoln.ac.uk/agents/server/, last visited 2003

[45] W3C Web Ontology Working Group
http://www.w3.org/2001/sw/WebOnt/, last visited 10th November 2003

Author Details

Manjula Patel

Manjula holds a PhD in computer graphics and animation from the University of Bath, as well as an MSc in systems design from the University of Manchester and a BSc (Hons) in computational science and economics from the University of Leeds. Prior to working at UKOLN, she worked as a post-doctoral research officer, lecturer and computer manager in the School of Mathematical Sciences at the University of Bath. Manjula has been working as part of the research and development team at UKOLN since August 1998. She has been involved in numerous EU projects. Her current research interests relate to digital libraries, metadata, ontology, digital repositories and virtual and augmented reality representation of cultural artefacts.

Monica Duke

Monica graduated in Pharmacy at the University of Malta, where she developed an interest in computer science. She was awarded an MSc in Information Processing at the University of York, joining UKOLN as a software developer in June 2000. As part of the research and development team she has worked on JISC-funded projects researching standards and developing tools to support resource discovery on the Web. She contributed to the software development effort in the agentcities.NET project. Monica has now moved to the Distributed Systems and Services team at UKOLN, supporting the development of the Resource Discovery Network http://www.rdn.ac.uk and the eBank UK project.