Int. J. Human-Computer Studies (2001) **54,** 155–184
doi:10.1006/ijhc.2000.0453
Available online at http://www.idealibrary.com on IDEAL®

Extracting focused knowledge from the semantic web

Louise Crow

Artificial Intelligence Group, School of Psychology, University of Nottingham, University Park, Nottingham, NG7 2RD, UK. email: lrc@psychology.nottingham.ac.uk.

Nigel Shadbolt

Department of Electronics and Computer Science, University of Southampton, Highfield, Southampton, SO17 1BJ, UK. email: nrs@ecs.soton.ac.uk.

(Received 5 October 2000 and accepted in revised form 31 October 2000)

Ontologies are increasingly being recognized as a critical component in making networked knowledge accessible. Software architectures which can assemble knowledge from networked sources coherently according to the requirements of a particular task or perspective will be at a premium in the next generation of web services. We argue that the ability to generate task-relevant ontologies efficiently and relate them to web resources will be essential for creating a machine-inferencable "semantic web". The Internet-based multi-agent problem solving (IMPS) architecture described here is designed to facilitate the retrieval, restructuring, integration and formalization of task-relevant ontological knowledge from the web. There are rich structured and semi-structured sources of knowledge available on the web that present implicit or explicit ontologies of domains. Knowledge-level models of tasks have an important role to play in extracting and structuring useful focused problem-solving knowledge from these web sources. IMPS uses a multi-agent architecture to combine these models with a selection of web knowledge extraction heuristics to provide clean syntactic integration of ontological knowledge from diverse sources and support a range of ontology merging operations at the semantic level. Whilst our specific aim is to enable on-line knowledge acquisition from web sources to support knowledge-based problem solving by a community of software agents encapsulating problem-sloving inferences, the techniques described here can be applied to more general task-based integration of knowledge from diverse web sources, and the provision of services such as the critical comparison, fusion, maintenance and update of both formal informal ontologies.

© 2001 Academic Press

KEYWORDS: ontological knowledge; software architecture; domain knowledge; semantic web; information integration; task models.

1. Introduction

The consensual terms and representations that make up ontologies are the lifeblood of domains of knowledge and expertize. The existence of the web has been a driving force in initiatives within the field of knowledge engineering for making such knowledge sharable and standardized. Well-formed Al ontologies, such as those that reside on servers like the Ontolingua Ontology Server (Farquhar, Fikes & Rice, 1996), are now available world wide. However, knowledge engineers are not the only people interested in sharing

 © 2001 Academic Press

knowledge. The Internet has catalysed an enormous global interest across communities in knowledge sharing and representation. Outside the AI ontology community, industry has been regularly using ontologies successfully for years (even though they may not call them ontologies) (Uschold & Jasper, 1999). Arguably, the most significant effect that the web has had is in making these conceptualizations (and resources of all kinds that are based on these conceptualizations) globally accessible and machine-readable. There are many such resources on the web that contain high-quality knowledge about domains. These are knowledge sources that have ontological qualities, by which we mean that they define concepts and relationships used in a domain, but are not formal AI ontologies. As we move closer to the vision of a "semantic web" in which knowledge is self-describing and machine readable (Berners-Lee, 1999), some of these knowledge sources possess explicit and defined semantics. Others do not, but still show a high level of structure and quality (McGuinness, Fikes, Rice &Wilder, 2000).

In this paper, we present an approach and demonstration architecture for extracting both explicit and implicit ontological information from web sources in order to create rich-structured domain ontologies on-the-fly. We believe that knowledge-level models of tasks have an important role to play in structuring and extracting useful and focused problem-solving knowledge from these web sources. There is great potential for tools which bring together knowledge engineering models and techniques for structuring knowledge and the large amounts of domain knowledge contained in web resources.

Components of domain knowledge—concepts and relationships—are common currency. This is the level at which normal task-based discourse occurs amongst domain experts. However, the analysis of problem-solving strategies and their domain knowledge requirements is an abstract challenge that requires careful analysis and is not frequently considered by those solving a problem in a domain. This challenge must be met in order to organize domain knowledge efficiently for problem solving. Our approach is informed by research from the knowledge engineering community that has established a canon of task types and associated problem-solving methods independent of the domain in which tasks appear. These task descriptions are used to filter web content in order to produce rich domain ontologies that are highly tailored to the knowledge requirements of the task in which they are to be used (see also Crow & Shadbolt, 1998, 1999).

We demonstrate the potential of the approach in the Internet multi-agent problem solving (IMPS) prototype. IMPS is a modular and extensible agent-based system that features the clean syntactic fusion of web content from heterogeneous sources. The complementary properties of these sources are exploited to create expanded enriched ontologies. This is accomplished using heuristic extraction and filtering techniques and the re-representation and fusion of web content in a flexible and expressive frame system. IMPS also supports knowledge editing and supplies the user with a range of operations for merging and reconciling ontological knowledge from multiple sources at the semantic level.

Our specific aim is to enable on-line knowledge acquisition from web sources to support knowledge-based problem solving by a community of software agents encapsulating problem solving inferences. We envisage knowledge engineering metatools that can be configured on-the-fly to acquire domain knowledge for a variety of tasks from the web and that will solve problems using the knowledge acquired. Such knowledge engineering tools would support every stage of the KBS development process and make

extensive use of web resources. However, the approach has broader potential to supply general tools which will automate the extraction of knowledge from the web using task requirements and provide critical knowledge fusion, editing and maintenance services for both formal and informal ontologies. These ontologies can facilitate the principled integration of knowledge resources on the web.

The structure of the rest of this paper is as follows. Section 2 outlines the roots of our approach in knowledge engineering. In Section 3, we narrow our focus to the construction of ontologies, examining the requirements for useful ontology tools. In Section 4, we discuss the potential for extracting ontological information from the web, considering three kinds of source—upper level ontologies, resources that use semantic markup and those with implicit structural information. In Section 5, we describe the IMPS prototype system in terms of functionality, architecture and implementation, illustrating this description with a scenario of the system in use, and in Section 6 we position the work with respect to other research in the field. Finally, in Section 7, we outline future directions in which the approach may be developed.

2. Knowledge engineering

In order to reproduce expert levels of performance in artificial systems, knowledge engineering has moved towards a paradigm of modelling expert behaviour. Investigations into human expert behaviour have revealed that there are some characteristics that are common across disciplines. Among these are the possession of a large amount of knowledge concerning the domain of expertize (e.g. Johnson *et al*., 1981; Gobet & Simon, 1995), and a superior and task-oriented mental representation of that knowledge (e.g. Chase & Simon, 1973; Chi, Feltovich & Glaser, 1981; Gobet & Simon, 1996). The difficulty of extracting, filtering and organizing knowledge from domain experts has challenged the knowledge engineering enterprise, producing a research community that is extremely interested in reusing knowledge wherever possible. The basic problem is one of extracting formal and consistent representations of knowledge that are suitable for inference from experts' expressions of knowledge, which may be informal and may contain operational knowledge only implicitly.

This problem has some striking similarities with the challenge being posed by the "semantic web" ideal in which large amounts of networked knowledge are made available for machine-assisted reasoning. In making sense of web content, the issue is one of transforming a large mass of relatively unstructured information (represented semi-formally for human-readability) into formal knowledge representations that can support machine inference and are sufficient for the knowledge requirements of specific tasks. Therefore, it is interesting to consider the mechanisms and practices that have been developed within knowledge engineering to deal with the challenges of the bottleneck of knowledge acquisition.

The current practice in knowledge engineering is to divide the knowledge involved in producing expert behaviour into a set of relatively independent models conceptualized at the "knowledge level" (Newell, 1982). These epistemological models can be grouped into task models, problem-solving methods and domain models. One of the major achievements of the field has been the development of a comprehensive set of descriptions of generic knowledge intensive tasks and problem-solving methods. These provide

vocabulary and models to express the abstract, domain-independent aspects of expert problem solving. This is important because the models can be reused across domains to direct knowledge acquisition from experts and structure domain knowledge efficiently for problem solving. Task models have been developed, refined and expressed to a level where they can be used to direct the knowledge acquisition process, specifying the necessary and sufficient domain knowledge for an application. In knowledge engineering methodologies, such as CommonKADS (Schreiber *et al*., 2000), the effort involved in building applications is reduced by fine-tuning an application model from one of a library of generic task types and applying an associated problem-solving method. The task model plays a mediating role between domain knowledge and problem-solving methods, allowing the domain-independent expression of the roles that domain knowledge can play in a task. It provides a flexible connection between domain knowledge and the inference structures that can be applied to that knowledge as its roles can then be linked to more than one problem-solving method.

The knowledge-structuring property of task models and PSMs that makes them an asset in knowledge acquisition has another aspect. The interaction problem (Bylander & Chandrasekaran, 1988), where both the content and representation of domain knowledge is affected by the nature of the task and problem-solving method, poses a real problem for the sharing and reuse of existing domain knowledge models for new tasks. The limits that the interaction problem places on domain knowledge model reuse (which we discuss further in Section 3), along with the problems of knowledge acquisition contribute to an interest in rapid and efficient tools and methods for domain model construction and a desire to use textual sources for domain knowledge acquisition. Increasingly, such domain knowledge resources can be found on the web (Section 4).

3. Ontologies—the state of the art

Recently, there has been increased interest in ontologies from the wider Internet community. Taxonomic subject ontologies such as the Yahoo! Index are in widespread use assisting browsing and web site search. Also, pioneering information integration systems such as Infosleuth (Nodine, Perry & Unruh, 1998), RETSINA (Sycara, 1999) and Ariadne (Knoblock *et al*., 1998) rely on common domain ontologies to collate knowledge for a user from diverse and distributed sources. The use of ontological models to access and integrate large knowledge repositories in a principled way has an enormous amount of potential to enrich and make accessible unprecedented amounts of knowledge for reasoning. This is illustrated particularly clearly in the medical sciences:

> "Most of the larger computerised resources in molecular biology, such as Genbank of the National Center for Biotechnology Information or Protein Data Bank (Bernstein *et al*., 1977) use databases rather than knowledge bases, and do not define an ontological framework other than the database schema. Each object has dozens of slots that describe where it comes from, what it consists of, its properties etc. However, there is no attempt to define general taxonomic, partonomic or role relationships among the concepts."
>
> (Hafner and Fridman Noy, 1996)

Within knowledge engineering, there is interest in the idea of reusing existing domain ontologies, and particularly in libraries of reusable domain ontologies to complement the

libraries of domain-independent knowledge components that are now available. A library of such reusable domain ontologies is definitely on the "wish list" of many KBS researchers. However, there are several significant challenges to the idea of ontology libraries in which whole ontologies or ontology components are stored for reuse. The sheer amount of knowledge that would have to be included in comprehensive library is one problem (van Heijst, Schreiber & Wielinga, 1997), and a large library brings with it indexing and accessibility concerns. Currently, the search for appropriate ontologies is presented as "hard, time consuming and usually fruitless" (Arpirez, Gomez-Perez, Lozano & Pinto, 1998). Another issue is a manifestation of the interaction problem—useful ontologies developed for an application are biased in terms of content and representation towards the requirements of that application. So, for example, an ontology describing a pair of lungs for the purposes of medical diagnosis might place emphasis on the description of symptoms of illness and possible fault states, whereas the representation of lungs in an ontology intended for use in the classification of different species might only represent those features which can be used to distinguish between animals. This may be a specific task bias or a more implicit reflection of the circumstances under which the original application was constructed (O'Leary, 1997). The re-representation of ontological knowledge at the symbol level is actually a practical impediment to reuse, particularly in attempts to use an ontology that has not been developed within the same metatool or framework (e.g. Cupit & Shadbolt, 1994). This may be one reason why there are still so few examples of existing ontology reuse in the literature (Uschold, King, Moralee & Zorgios, 1998*b*).

Conceptual change and disagreement amongst participants are also problems for the notion of ontology libraries. Usually, the knowledge in knowledge-based systems represents the "state of the art" in some specialist branch of human expertize. Although obsolescence is more of a problem with instantiated knowledge bases, domain conceptualizations also change as disciplines advance, and ontologies will need updating. Stable ontologies do occur in some fields through standardization efforts and/or years of conceptual development (e.g. the Periodic Table in the physical sciences). It may be the case that such an ontology is sufficient for new applications. Alternatively, the problems being tackled may require some modifications or additions to existing representations. Further, current experts can differ both in the concepts they use and the terminology they use for the concepts (Shaw & Gaines, 1989).

One way to control these biases is to attempt to store smaller components of ontologies and to make as many assumptions as possible explicit when the ontology is originally constructed. Two major practical ontology library projects have both taken this approach—the KACTUS (Laresgoiti, Anjewierden, Bernaras, Corera, Schreiber & Wielinga, 1996) project for reusable ontologies for technical systems, and the GAMES-II project in the medical domain (van Heijst, *et al*., 1997). GAMES-II tackles the interaction problem with a hierarchical component-based library which attaches "method-specificity" and "domain-specificity" attributes to concepts which get more specialized as the hierarchy is descended. There is a limit, however, to the amount of diversity that can be represented before knowledge becomes too fragmented and contradictory to deal with. O'Leary (1997) introduces "paradoxical lack of reuse"—the idea that in situations and domains that facilitate ontology reuse (i.e. those that have low change, high consensus, and are well structured), these same characteristics ease development

of ontologies on a bespoke basis and therefore decrease the need for reusable ontologies.

Whichever approach is taken to the construction of ontology libraries, ontology reuse in practice seems to actually imply assembling, translating, extending, specializing and adapting exisiting ontologies (see, for example, Uschold, Clark, Healy, Williamsons Woods, 1998*a*; Russ, Valente, Mac Gregor & Swartout, 1999). We argue that selecting task-relevant domain knowledge and mapping it to task requirements will always be a significant hidden cost in ontology reuse (even if libraries of ontologies are widely available) and is very far from being automated. This cost must include all the work needed to find, understand and adapt pre-existing knowledge for reuse. In component-based ontology reuse scenarios, mapping needs to occur not only between domain ontology, task and problem-solving method, but between diverse components that are assembled together to make up the ontology. In order to be used most efficiently, component-based ontology libraries must be supported by more powerful ontology manipulation and integration tools which support the processes that libraries require. Such tools would include component brokers (e.g. $(ONTO)^2$ Agent (Arpirez *et al*., 1998)) and also ontology manipulation tools that support a wide range of operations for constructing ontologies using available knowledge. These latter tools would combine operations for the integration, translation and development of existing ontologies with support for generating new concepts and relationships as required. We present IMPS as such a tool. Given that what we know as formal AI ontologies may require symbol level re-representation and significant semantic reconciliation before they can be merged, we argue that the capabilities required to these tools can be usefully extended to allow the filtering, manipulation and exploitation of the huge resource of less formal ontological knowledge present in web sources.

4. Ontological knowledge on the web

By using web sources, we open up the possibility of rapid ontology formation over a huge range of domains for a variety of purposes. Within knowledge engineering, the bottleneck of knowledge acquisition has led to the use of textual sources, particularly in the first stages of knowledge acquisition. The representations of domain knowledge available on-line could be used to ease the KA bottleneck by allowing the rapid construction of a domain model on which to base the elicitation of expert knowledge. However, the use of web sources does place some addition requirements on an ontology construction tool.

Whilst ontology components in a library may be well formed and even indexed according to task and method, the construction of domain ontologies using web sources will entail the identification and extraction of the correct task-relevant knowledge. We propose that the strong conceptualizations of tasks and methods and their domain knowledge requirements developed in knowledge engineering methodologies can be used to direct this extraction. Further, the use of sources which may not be representationally expressive or formal require a tool to supply operations for the formalization and refinement of knowledge and knowledge quality assessment. Web sources are open to change in knowledge representation methods and information presentation technology. So a system using them must also be to some degree open-ended in order to deal with emerging standards in knowledge representation and information delivery. In the

next sections, we will outline the potential of three different sources of domain ontologies on the web—explicit semantic markup languages, upper level ontologies, and sources with implicit structural ontologies.

4.1. SEMANTICS ON THE WEB

Semantic markup offers the potential to allow applications to access the semantics of web resources. We believe this is a significant step in making automatic knowledge acquisition from and coherent inference over web repositories possible. There have been several attempts within the knowledge engineering community to define extensions and supplements to HTML, the existing standard for web knowledge representation, to allow a richer representation for expressing the relationships between concepts that structure domains. For example, the KA^2 initiative (Benjamins & Fensel, 1998) and SHOE project (Simple HTML Ontology Extensions) (Luke, Spector, Rager & Hendler, 1997) shared the notion of re-organizing the Internet into a richer structure using some kind of meta-content format enabling documents on the web to describe themselves.

More recently, Extensible Markup Language (XML) (Bray, Paoli & Sperberg-McQueen, 1998) has emerged as a richer general representation standard for semi-structured knowledge and is being used to markup documents on the web. XML differs from HTML in three main ways. It is extensible—users can create their own domain-specific tags or attributes to semantically qualify their data. It also allows for the specification of deep structures. Finally, XML has a document type definition (DTD) that allows users to define the tags they have created. This forms a description of the XML grammar for use by applications that need to validate a document as being of a certain sort.

In effect, this DTD expresses an agreed set of terms and relationships that anyone writing an XML document using the same DTD must comply with. It is easy to consider a DTD as, in effect, a simple domain ontology. However, Fensel (2000) asserts four differences between XML DTDs and ontologies—(1) a DTD specifies the legal *lexical* nesting of a document, which may or may not coincide with an *ontological* hierarchy and as a result; (2) DTDs do not have any notion of inheritance; (3) DTDs do not have very rich typing concepts—a tag can be either composed of other tags, or it can be a string; and (4) DTDs define the order in which tags appear in a document. In an ontology the ordering of attribute descriptions does not matter. We argue that these are essentially symbol level differences in the expressiveness of the knowledge representation that a DTD provides and that nevertheless, a DTD is a rich source of ontological knowledge for the domain that it describes, having been generated by those working in the domain to represent knowledge about the terms and relationships that they need to communicate. It is also worth remembering that XML and associated DTDs are part of the first generation of widespread meta-data on the web, and as such represent only a fraction of the power of representation that may be available in the future.

There are two large-scale initiatives being launched for an expressive standard semantic markup language for the web. Both rely on XML. One is the DARPA Agent Markup Language (DAML) (Hendler, 2000; Rapoza, 2000), and the other is the Ontology

Inference Layer (OIL) (Fensel, Horrocks, Van Harmelen, Decker, Erdmann & Klein, 2000). OIL aims to provide a web-based representation and inference layer for ontologies. It combines modelling primitives from frame-based languages with the formal semantics and reasoning services of description logics. The initial application of OIL is in knowledge management in large and distributed organizations—the On-to-Knowledge project specifies that unstructured and semi-structured data will be annotated automatically using OIL, and agent-based user interface techniques and visualization tools will help the user navigate and query the information space. DAML is a language developed under the asupices of DARPA aimed at representing semantic relations in machine readable ways to allow agents to conduct information fusion from diverse sources and intelligent search. DAML will allow objects in the web to be marked up to include descriptions of the information they encode, descriptions of the functions they provide and/or descriptions of the data they can produce.

4.2. UPPER-LEVEL ONTOLOGIES

The availability of high level, very broad coverage ontologies on the Internet is increasing. Some of these ontologies, such as the "upper Cyc ontology" (Lenat, 1995) are explicitly aimed at supporting reasoning. Others, such as the generalized upper model (Bateman, Magnine & Rinaldi 1994) and WordNet (Miller *et al*., 1993), are high-level linguistic ontologies. It has been argued that linguistic ontologies represent the "world view" encoded in natural language and thus have a lot to offer as seed ontologies for specialization to be used in domain ontology formation (Dahlgren, 1995; Guarino, 1996). We believe that these could be particularly useful in bridging the gap between "common sense" knowledge and the domain-specific knowledge which can be obtained from specialist sources by providing a general high-level structure in which to situate domain-specific knowledge. One specific advantage of using a linguistic source is that it is easily understood by someone who is not a domain expert, being grounded in the common shared understanding of terms. It is also very keyed in to differences in word meaning, so this information is relatively easy to extract.

Grounding ontology construction in an upper-level model is particularly useful when merging domain-specific ontologies or ontology components. When disjoint groups merge ontologies the meaning of terms can lose their context, and can differ—this effect is at its most insidious when the terms only differ slightly (Schuyler, Hole, Tuttle & Sheretz, 1993; Wiederhold & Jannink, 1999). Building ontologies downwards from a high-level ontology allows differing ontologies to be compared within the same broad conceptual framework. Weinstein and Birmingham (1999) put forward a set of algorithms for determining measures of "description compatibility" in differentiated ontologies. Their ability to do this, they claim, rests on the following assumptions. First, that a formal framework that supports subsumption is used for relating concepts to one another; and secondly that the concepts being compared inherit definitional structure from concepts in shared ontologies. Measures of compatibility can be seen as a first steps towards reconciling different ontological viewpoints. This upper-level approach has been pursued in several practical ontology construction projects (e.g. Swartout *et al*., 1996; Gangemi, Steve & Giacomelli, 1996; Pisanelli, Gangemi & Steve, 1999).

4.3. IMPLICIT STRUCTURAL ONTOLOGIES

Although the use of semantic markup is becoming more common, self-describing resources are still in the minority on the web of today and may be so for some time to come. However, much of the domain information on the web is structured in a meaningful way, such as that held in databases. Although it is not explicitly represented, the structure of these sources contains implicit information about the conceptualization of the domain that they express. There are several indications in the literature that this could be a potentially useful source of ontological information. van Harmelen and Fensel (1999) point out that machine-accessible representation of the semantics of on-line information sources can be achieved using semantic markup or by writing programs to procedurally extract the semantics of web sources, such as filter, wrapper and extraction programs, and furthermore that these approaches are complementary. We propose an architecture which combines the ability to use explicit semantic markup schemes where available with the active extraction of implicit ontological structure from sources which are less expressive. Simple heuristics and combinations of extraction techniques can be used to infer domain structure from such sources.

5. The IMPS system

The IMPS architecture is designed to facilitate the retrieval, restructuring and fusion of information from the web according to (1) the domain and (2) the kind of task in which the information is to be used. In the following sections we will discuss what the system does in more detail and how the architecture and implementation support this functionality.

5.1. FUNCTIONALITY

The IMPS architecture is ultimately designed to provide both knowledge acquisition and inference components encapsulated in agent shells. In order to realize and support such an architecture, the implemented prototype focuses on the problem of extracting task- and domain-focused knowledge from the web. The first stage in this process is the generation of ontologies to structure that knowledge. We assume that the nature and content of the ontology used in problem solving will be affected by the task and the method being used. A second assumption is that the domain knowledge requirements of well-specified task models and problem-solving methods can be used to structure this ontology from the outset, and that this will reduce the overhead of mapping between the task model, problem-solving method and the ontology. CommonKADS specifications of the knowledge requirements of tasks and methods are used to drive ontology construction. For example, in order to support typical inferences performed in classification tasks [Figure 1(a)], domain knowledge must describe the object type to be classified, and potential classes into which an object of that type can be grouped [Figure 1(b)]. By linking domain keywords supplied by the user to knowledge roles in the task, the system can compose hierarchies of concepts and appropriate relationships around those keywords [Figure 1(c)]. In this way, IMPS acts as a "concept multiplier" (Swartout *et al.*, 1996), speeding up ontology construction by coupling user–supplied knowledge with

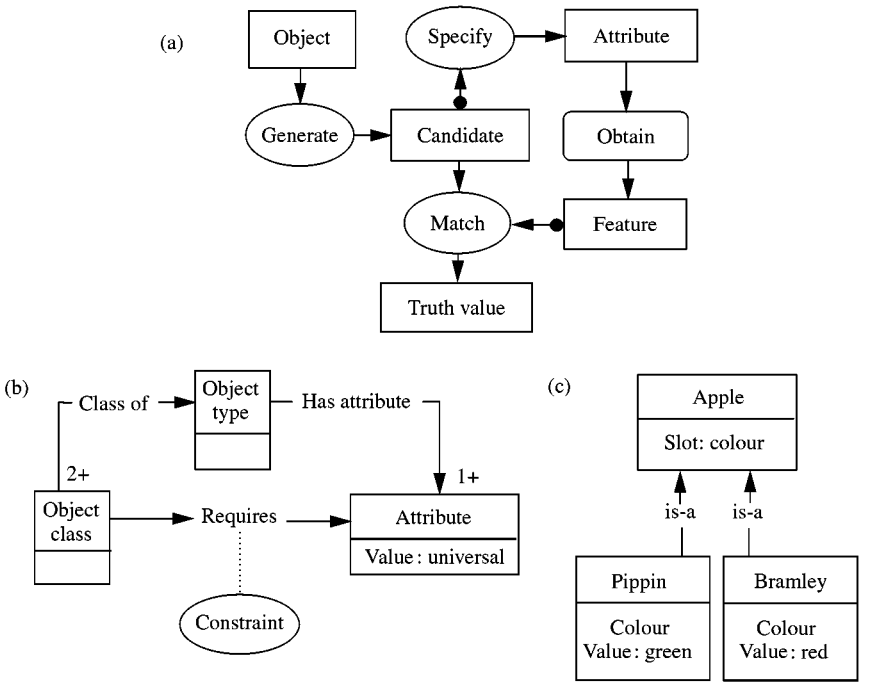


FIGURE 1. (a) Inference structure for the pruning classification method (Schreiber *et al.*, 2000). (b) Typical domain knowledge schema for classification tasks (Schreiber *et al.*, 2000). (c) An example of domain knowledge structured according to the requirements of the domain knowledge schema.

large information resources and simple heuristics to perform rapid ontology formation. The usage lifecycle of IMPS is linear, comprising five stages.

5.1.1. Initialization. During initialization, the user primes the IMPS system with the kind of task that is to be attempted and an indication of the domain in which the task is situated. As the knowledge engineer will not have a clear model of the domain at this stage, this is a simple domain keyword. The user is asked to match the domain keyword they have supplied with a knowledge role in the task model. This produces an initial mapping from the task model to the domain term. The system can then develop knowledge structures around the domain term so that the ontology fulfils the knowledge requirements of that role in the task model. The final piece of information required is the location of Internet-accessible knowledge sources that will be used to create the ontology.

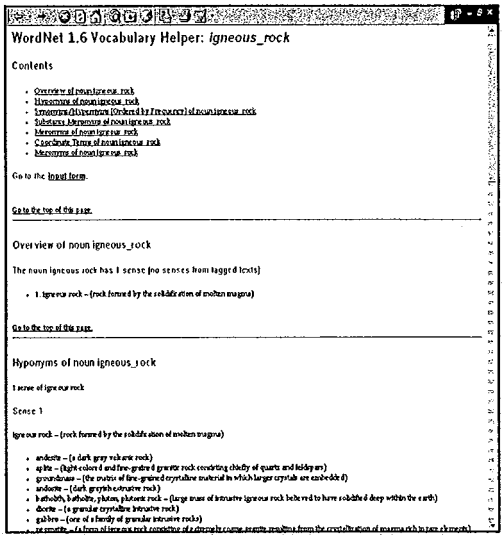
5.1.2. Disambiguation. Many words are polysemous and the meaning of terms can change according to the domain in which they are being used. In order to develop an ontology around the correct concepts, the system must resolve any ambiguity about the meaning of the domain keyword given by the user. This is accomplished by using a high-level linguistically oriented knowledge source to provide the user with definitions of the term. The user can choose the one that matches the concept they want to develop.

5.1.3. Primary source extraction. Once the initial concept has been identified, the system extracts knowledge from the primary knowledge source to build a structure of related concepts around it. In order to do this, the task model must be consulted to determine which structuring relationship between concepts is critical for inference with relation to this role in the task model, and which other relationships and aspects are important. Matches are made between the kinds of knowledge required and the knowledge available from the source.

The upper-level ontology used for this stage in the prototype is WordNet, a semantically orgnaized lexical database. It is one of the most well-developed lexical ontologies in existence (Fridman Noy & Hafner, 1997). English nouns, verbs, adjectives and adverbs are organized into synonym sets (synsets), each representing an underlying lexical concept. These synsets are linked by different relations such as synonymy, hyponymy and meronymy. A query to the WordNet web interface (Peterson, 1999) for information on a word will produce an HTML page. This shows the set of synonyms (synset) the word belongs to, with a definition of this concept, plus lists of other synsets and definitions, grouped according to their relationship to the term submitted. The HTML pages that are presented are semi-structured—they combine natural language text content with consistent formatting, making them highly accessible to rule-based information extraction. The IMPS extraction rules combine syntactic and semantic constraints with delimiters that "bound" the text to be extracted. These segment documents so that patterns can be applied only to the relevant pieces of text. Together, these heuristics are used to extract concept information that is translated into the IMPS frame representation, producing a hierarchy of semantically related concept frames. Heuristics are used to infer possible slots on these concept frames. The structuring hierarchy is organized by hyponymy (is-a) or meronymy (has-part) relationships depending on the requirements of the role and task model.

The primary source is not domain specific, and so it is not expected to have a very sophisticated model of any one domain. It may contain serious omissions and incorrect information. Therefore, some error checking mechanisms are used to identify and highlight potential semantic inconsistencies using the structure of the information extracted (see Crow, 2000 for details). The objects and relations extracted from the general source are presented back to the user graphically in a simple, browsable form as a skeletal hierarchical ontology for the domain. The representation of concepts includes definitions where available and relationships to other concepts in the ontology (Figure 2). At this stage, the user can add and remove concepts from the hierarchy. They can also edit the properties of concepts and change their relationship to other concepts.

5.1.4. Secondary source extraction. Once the user has edited the skeletal ontology, IMPS extracts concepts from the second source. This is a domain-specific source, which will probably have been authored by someone involved in the domain. This means that it has slightly different attributes from a general resource. It may require user navigation and intervention, and the relationships used to join concepts may not be very expressive or refined—as domain experts, the authors may be less interested in knowledge representation issues. It may also be harder to automatically determine what kind of information may be available. Therefore, the knowledge is translated into the same representation format as the skeletal ontology, retaining as much of the information contained in the



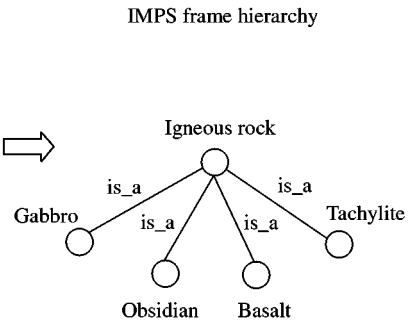


FIGURE 2. Extraction of a high-level skeletal ontology in the domain of geology using WordNet.

original source as possible, but IMPS interprets ambiguous or implicit relationships between concepts extracted from this source in task-relevant ways. The secondary sources currently available to IMPS are XML DTD documents and plaintext multiple table databases.

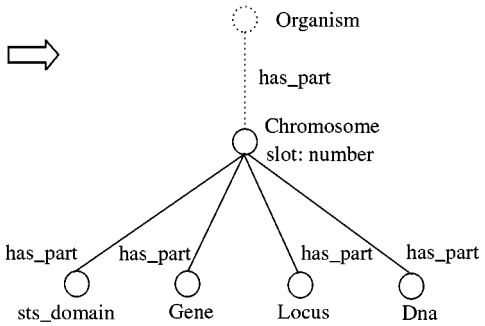
In order to extract concepts and relationships from XML DTDs, IMPS uses an XML parser (Mayuyama, Tamura & Uramoto, 1999) to parse their contents. This works with the document object model (Level 1), which provides a standard set of objects for representing HTML and XML documents, a standard model of how these objects can be combined, and a standard interface for accessing and manipulating them.

IMPS extracts elements and attributes from the DTD, transforming elements into concept frames and attributes into slots on those concept frames (with possible values if available from the DTD) (Figure 3). It translates the relationships between nested elements into hyponymy or meronymy relationships between concept frames, depending on the relationship that is being used to structure the ontology. XML DTDs use a relatively inexpressive representation format in which the nesting of concepts can be used and interpreted in a number of ways. For example, it can be used to specify properties of a concept or parts of the concept. As the syntax gives no clue to the precise semantics it encodes, IMPS translates the document to produce the relationships that are most important for the task model being used. It also allows the user easily to change the relationships between concepts and the way they are are expressed.

In extracting information from databases, IMPS reads in a set of HTML files describing the layout of the tables of a plaintext database. The user is asked to choose a table to use for extraction, based on a summary of their contents. The relevant layout page is processed and the information it contains about the syntactic structure of the database table (e.g. column widths and titles, URL of the table) is used to parse the table.

(a)

```
<?xml encoding='UTF-8' ?>
[...]
<!ELEMENT chromosome  (#PCDATA |
sts_domain | dna | locus | gene |
%nr; )*>
<!ATTLIST chromosome  %global;
 number CDATA  #IMPLIED >
```



(b)

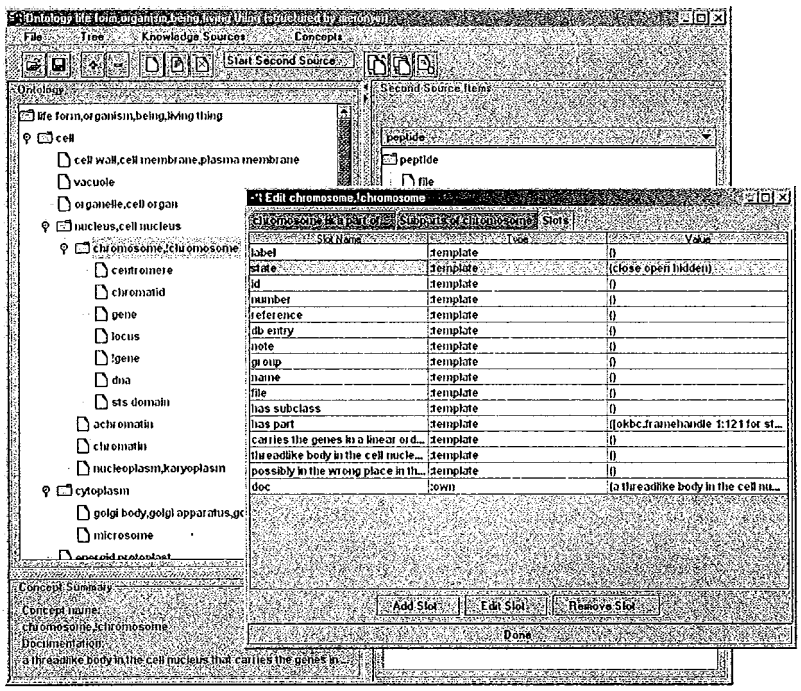
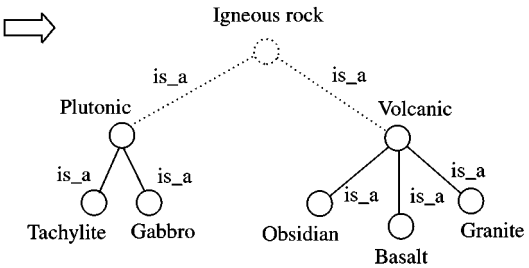


FIGURE 3. (a), (b) An ontology fragment in the domain of genetics is generated from the structure of an XML DTD.

A simple heuristic is used to convert the information into concept trees. The class identifies "instance" columns in a database in which each row has a different value (such as the "Rock name" column in the table shown in Figure 4, and "class" columns, in which values are repeated (such as the "Rock-type" column). The heuristic used is that, as there is a one-to-many mapping between items in the unique columns and items in the classification columns, the unique values from the first columns can be interpreted as

(a)

Rock name	Rock type
Tachylite	Volcanic
Basalt	Volcanic
Gabbro	Plutonic
Obsidian	Volcanic
Granite	Plutonic



(b)

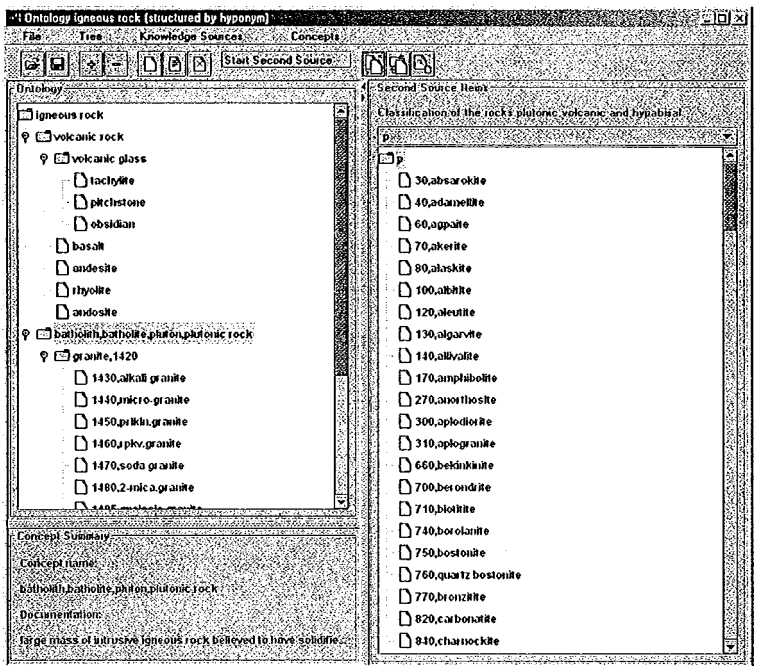


FIGURE 4. An ontology fragment in the domain of geology is generated from the structure of a database table.

hyponyms (subclasses) of the items in the "class" columns. Alternatively, if the structuring relationship being used is meronymy, the unique values can be interpreted as parts of the classes.

5.1.5. Concept matching and integration. The concept hierarchies that the system has developed using the specialist source are presented to the user alongside the skeletal ontology. As a complement to the automatic import and restructuring of knowledge, the user is supported in adding concepts into the ontology in a variety of ways. IMPS is

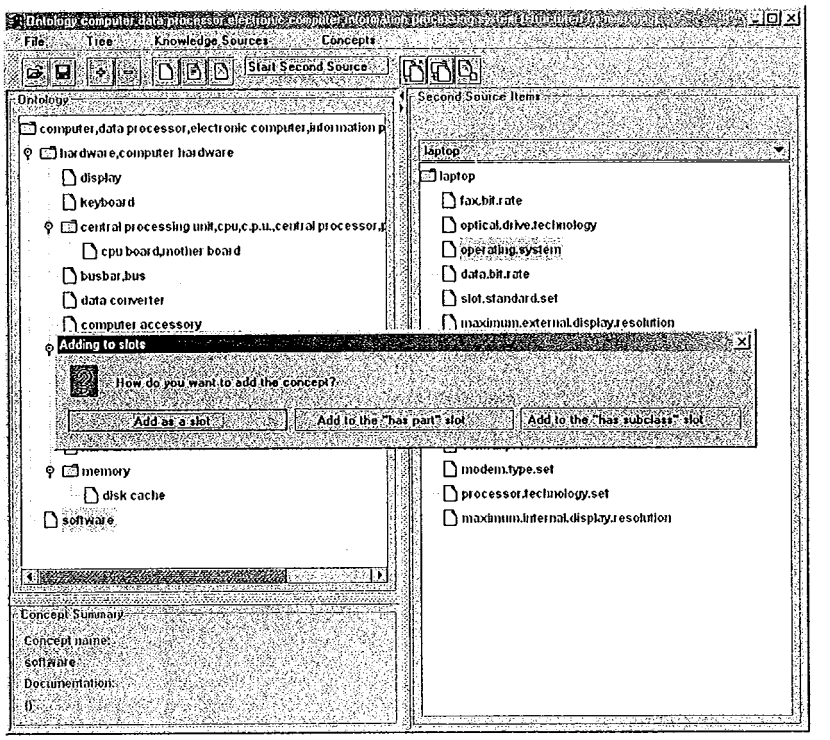


FIGURE 5. Several operations are supplied for changing the way concepts are represented.

specifically required to support the user in re-representing, formalizing and integrating knowledge from multiple sources that may not only express different domain perspectives, but may also vary in their level of expressiveness. This necessitates operations that allow the user to freely change the way in which concepts are represented, for example merging a concept that was represented as a class frame into the ontology as a slot on an existing class (see Figure 5).

IMPS, therefore, supports the following as basic operations for merging concepts from two sources.

- Merge(F1, F2)—Merges two frames, F1 and F2, according to the structuring relation being used in the ontology. Names are merged, slots and values are tansferred to the new frame, and relationships to other frames are inherited.
- Add as Part(F1, F2)—Adds frame F2 into the ontology, linking it to fame F1 using a "part-of" relationship.
- Add as Subclass(F1, F2)—Adds frame F2 into the ontology, making it a subclass of frame F1.
- Add as Slot(F1, F2)—Adds frame F2 into the ontology, making it a slot on frame F1.

Within the ontology, IMPS supports standard operations for manipulating of the class hierarchy, such as adding superclasses and subclasses to a class. In a partonomy, it also

provides one step operations which simplify the manipulation of meronymic "part-of" relationships between concepts:

- Get Parts(F)—Returns a list of all the frames that are linked to frame F by a "part-of" relationship.
- Get Wholes(F)—Returns a list of all the frames that frame F is linked to by a "part-of " relationship.
- Add Part(F1, F2)—Links the frame F2 to frame F1 using a "part-of" relationship.
- Remove Part(F1, F2)—Removes the "part-of" relationship between frame F2 and frame F1.

Typical ontology construction operations, such as making a new class or slot, adding a value to an existing slot or changing the names of classes and slots are also supported.

5.2. ARCHITECTURE AND IMPLEMENTATION

In this section, we will describe the way in which the IMPS architecture supports the functionality we have described. IMPS is a multi-agent architecture, in which two specialist agents work together with the user, using a knowledge library which contains task model descriptions and knowledge extraction heuristics.

We believe that a multi-agent system is a suitable encompassing model for a knowledge engineering metatool that covers ontology construction, KA and inference using web knowledge sources. For more details on the agent architecture, see Crow (2000) and Crow and Shadbolt (2000). In order to meet the requirements of a specification which calls for appropriate and timely event-driven behaviour, but also a high level of intentionality and sophisticated symbol manipulation, IMPS uses hybrid agents built in two layers. IMPS is coded in Java, making use of the robust and easy networking functionality and object orientation. The basic structure on which the reactive layer of the IMPS agents is based is supplied by the Java Agent Template (JAT) 0.3 (Frost, 1996). This layer supports low-level behaviour driven principally by external events, such as instructions from the user and messages from other agents. On top of this, the deliberative layer supports more complex processing, prompted by changes in the internal state of the agent's knowledge base. To provide the deliberative layer in IMPS agents, the JAT is supplemented with Jess. Jess is a version of the popular expert system shell CLIPS, rewritten entirely in Java (Friedman-Hill, 1998). It provides the agents with internal representation and inference mechanisms. In effect, the addition of Jess means that whilst the agents share a common architecture, each agent reasons and acts like a small knowledge-based system following its own set of rules. Jess can be used to manipulate external Java objects so there is a tight coupling between the inference engine and the agent shell.

The agents use the knowledge query and manipulation language (KQML) (Finin, Labrou & Mayfield, 1997) to communicate, as specified and supported by the JAT, with KIF (Genesereth & Fikes, 1992) as the content language. The agents in the prototype are the following:

- The knowledge extraction agent (KExA), acting as an Agent Name Server (ANS) and the interface through which the user interacts with IMPS during initialization. The

KExA helps the user fill task templates with domain terms that can then be used by the OCA for ontology structuring.
- The ontology construction agent (OCA), which is able to use modules from the knowledge library to extract information from networked knowledge sources and re-represent it so that it can be integrated into a coherent whole.

The knowledge library component of IMPS is as essential to its operation as the agent component. The extraction classes used to obtain particular kinds of knowledge from knowledge sources are all based around common interfaces for general and specific sources, with standard inputs and outputs. The actual mechanisms by which the class extracts information from a source and parses it into a form comprehensible to the inference engine are completely hidden from the agent loading the class. New classes can be added to the library as appropriate, in a "plug-and-play" manner, without any change to the rest of the architecture. Within the library, the knowledge sources are indexed by type—e.g. database, XML, DTD or WordNet, etc., so new instances of a particular type of source just need to be identified as such to be used by the system. This is also true of the task model components, which are based around a (different) common interface. The system has four task models at present—classification, diagnosis, configuration design and assessment.

IMPS uses the open knowledge base connectivity (OKBC) knowledge model (in the Java implementation and the Tuple KB representation system binding) to represent the ontologies created (Chaudri, Farquhar, Fikes, Karp & Rice, 1998). The OKBC knowledge model is particularly well suited for fulfilling the requirement that knowledge should be represented both formally and informally, and translated from different sources showing various levels of expressiveness. It was designed to trade-off expressiveness and generality so that bindings could be specified from OKBC to representations as simple as a file system or as expressive as a theorem prover supporting full first-order logic. OKBC complies to current knowledge-sharing standards to allow the export of ontologies to other systems, and also supports fast and efficient querying and manipulation of the knowledge base.

The IMPS ontology editing interface is constrained by two major requirements. It must provide clear browsable visualizations of large ontologies, and it must support merging and editing operations between the ontology and ontological concepts from another knowledge source. These two requirements make it suitable for the application of the "focus + context" information interface paradigm (e.g. Lamping, Rao & Pirolli, 1995) which attempts to support both a close focus on specific concepts and the visualization of a concept node surrounded by the entire data structure representation of the concept space to show how a concept fits into the "big picture" (Korn, 1996).

There are two panes in IMPS that support ontology viewing and editing. The context pane is always available. It shows the ontology as a nested, indented rooted tree structure created using Java's Swing component set for interfaces. The representation of this tree is widget-based, like a file structure browser, rather than richly graphical. The user can expand and contract individual branches of the tree by clicking on them. Ontology editing and merging commands are available as buttons and menu items, with redundancy between the two. The concept operands for these commands are selected by clicking on them. Selected operands are highlighted, and only one concept for each knowledge

source may be highlighted (i.e. one concept in the main ontology, and one from the specialist source). In order to support the merging operations, the ontology and the ontological fragments to be merged are viewed on the same screen.

In order to complement the overview pane and provide the focus necessary to examine and edit concepts in detail, a second pane is presented when a concept is selected and the "Edit Concept" button clicked. This is a tabbed pane with three options. Two of these allow editing of the concept's super- and sub-class relationships in a taxonomy, or viewing and editing of the concept's "part-of" and "has-part" relationships in a partonomy, depending on which relationship is being used to structure concepts. The third allows the user to view and edit the concept's slots and slot values which are not visible in the overview pane.

5.3. SCENARIO OF USE

In this section, we illustrate our description of IMPS' functionality with a scenario demonstrating the potential of the system in use. Whilst the scenario has been constructed to display the current capability and future potential of the architecture, conveying optimal performance with an experienced user, it is completely grounded in the current capabilities of the tool. In the scenario, IMPS is being used by a knowledge engineer who is particularly interested in developing a knowledge model of genetic diseases to be used in a system for diagnosis. He uses IMPS to find out if this is feasible and explore ideas about the domain knowledge that would be required for this application. He has done some preliminary reading, but is not sure about the best way to represent the necessary knowledge.

5.3.1. Visualizing alternative ontology structures. The knowledge engineer considers first a genetic model that will support differentiation of the genetic characteristics of diseases—this will structure the most detailed information in the system. He asks IMPS to create a concept hierarchy structured by meronymy, using the keyword "organism" as its root (see Figure 6). He does this in order to produce a structural model that will support the specification of different genetic states. He trims some extraneous concepts, focusing on the cellular-level decomposition.

The knowledge engineer is aware of two proposed XML standards for representing knowledge about genetic sequences: Biopolymer Markup Language (BioML) (Fenyo, 1999) and Bioinformatic Sequence Markup Language (BSML) (Visual Genomics, 1999). He wants to look at both of these to see if they could contribute something to his ontology. First, he uses the DTD of BSML to see what it has to offer. Browsing through the concepts available, he finds "sequence" and decides to make this a sub-part of gene to represent sequences of DNA within a gene, knowing that the system using this ontology will need to represent the presence or absence of particular sequences of DNA as indications of disease mutations. Having done this, he examines the concept to see what attributes it has. These attributes, represented as slots, have been automatically created by IMPS from "attribute" statements in the source DTD (Figure 7).

He sees that the "sequence" concept in the BSML representation is a rich one, with many slots, allowing representation of attributes like the molecules that the sequence is made up of and the gross topology of the sequence. IMPS has also imported allowed

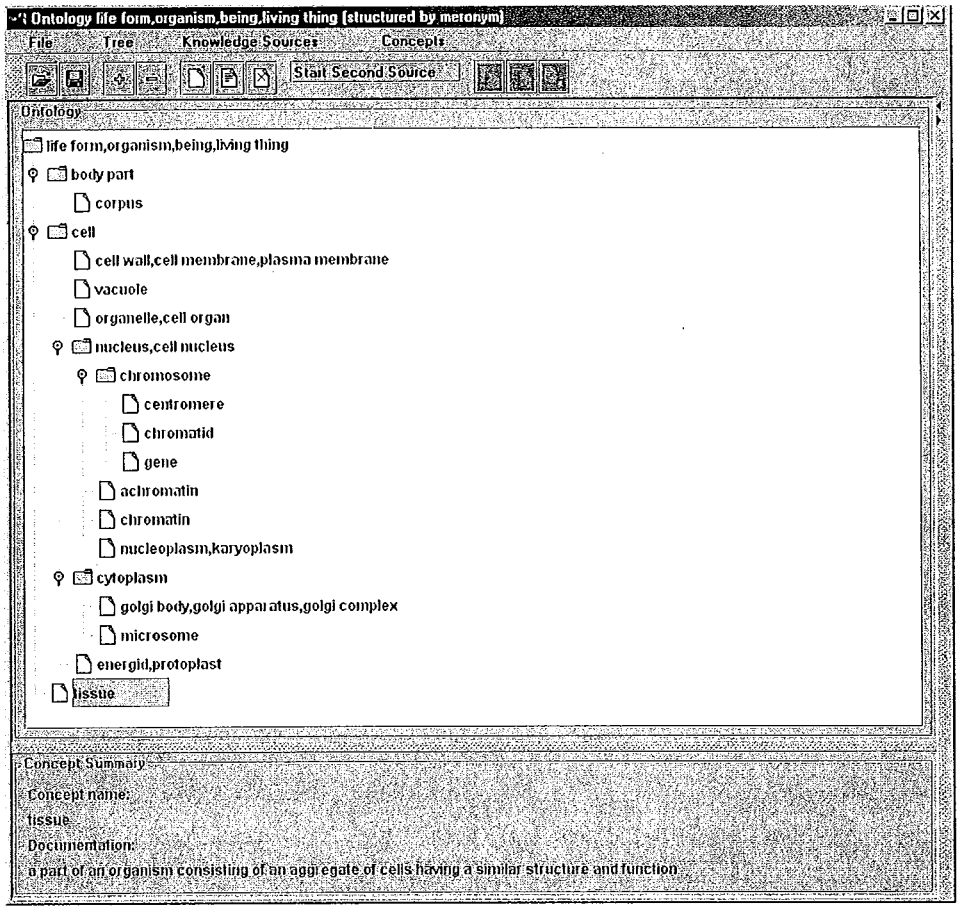


FIGURE 6. A concept partonomy rooted in the concept "organism".

value sets for some of these slots. However, he finds little else of interest to him in the BSML knowledge source, and decides to look at BioML as well.

5.3.2. Formalizing relationships between concepts. He starts his examination of the knowledge extracted from BioML by looking at high-level concepts that describe the location of a particular piece of DNA within an organism's compliment of chromosomes. He thinks that this information will be relevant for the diagnostic process. He considers merging the "chromosome" concept from BioML with the "chromosome" concept extracted from WordNet. He notes, though, that the BioML representation of "chromosome" has parts like "reference" and "file" which may be useful parts of the concept representation, but are not meronyms in the physical sense—they are not parts of a chromosome. The relationships between tagged concepts are described in the BioML documentation as expressing that an entity *belongs to* another entity. The knowledge engineer wishes to divide this rather vague notion of *belonging to* into concept properties

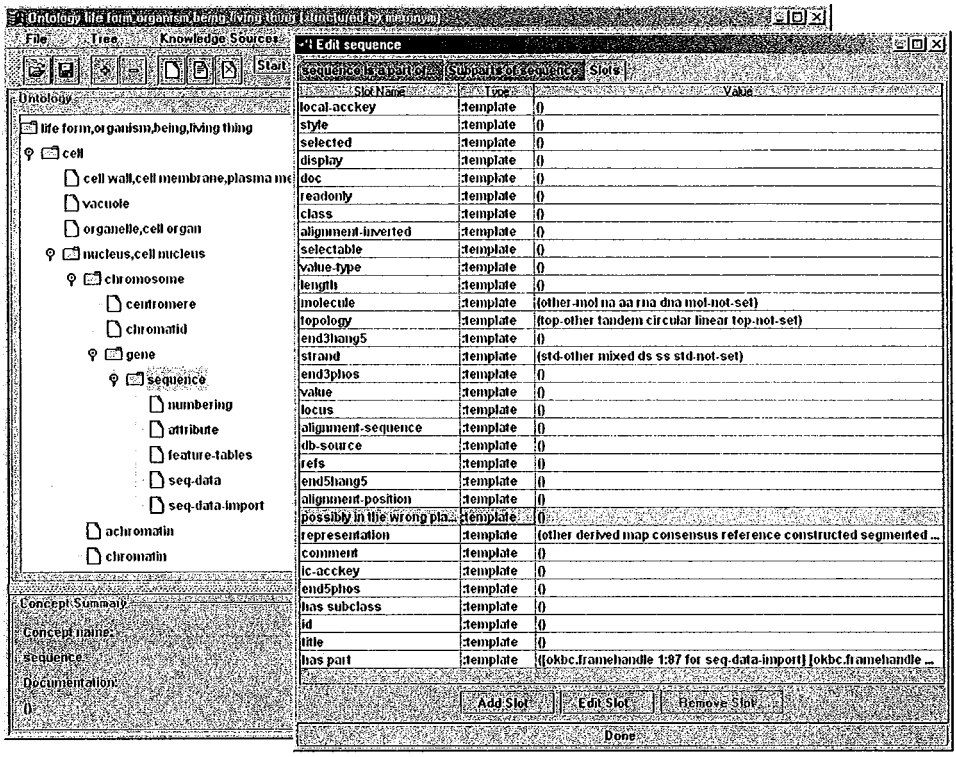


FIGURE 7. The concept "sequence", merged from the secondary source BSML DTD, has many automatically created slots and values.

and the physical parts of the entity the concept describes for the purposes of his application. He adds the abstract concepts that relate to references in the literature for the chromosome as slots on the concept to be filled with the relevant information if required, instead of maintaining them as "parts of" the concept. He then merges the two chromosome concepts (Figure 8).

As with BSML, IMPS has automatically added some slots and allowed values to the frame representation of the concept using information obtained from the BioML DTD. The knowledge engineer merges the "gene" concept from BioML with the "gene" concept in the main ontology and adds the concept "dna" as a part of "gene". He uses the BioML documentation to provide textual definitions of the concepts that have been added and to create slots and values. He is not totally satisfied that either the BioML or the BSML knowledge structures express the genetic knowledge that will be required for his diagnosis task, and so decides to investigate other parts of the ontology to see if he can clarify what the requirements on this part will be.

5.3.3. Composing ontologies to support multiple inferences. Firstly, the knowledge engineer knows that knowledge structures are required that will link diseases to symptoms and

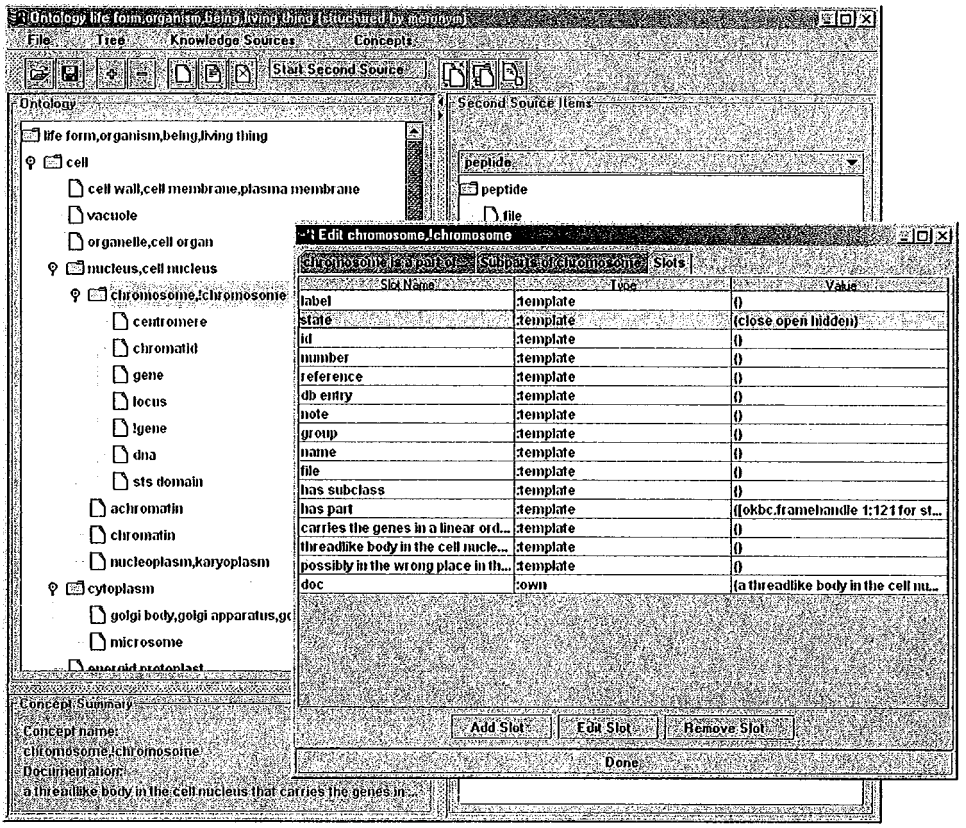


FIGURE 8. The concept "chromosome", merged from the secondary source BioML DTD.

also express knowledge about the prevalence of diseases. Domain knowledge will also be required to link these diseases to genetic characteristics. In order to situate this knowledge, he uses a hyponymy hierarchy that IMPS generates from the keyword 'genetic disease" using WordNet (Figure 9). This shows both specific diseases and disease classifications. He notes that diseases are sometimes classified by large grain genetic characteristics—two major categories of genetic disease "autosomal dominant disease" and "autosomal recessive disease" are characterized by mutated genes on a particular chromosome.

He sees that it will be important to represent chromosomes and their subtypes clearly in the genetic part of the ontology, and also that these classifications may be used to make the reasoning process more economical by ruling out classes of disease with a single observation. This sways him towards using BioML as a template, and potentially using it for knowledge acquisition to instantiate his ontology later, as it has more detail at the chromosome level. He remembers that he must check that the sub-types of chromosome such as autosome are present in the ontology. Editing the concept "genetic

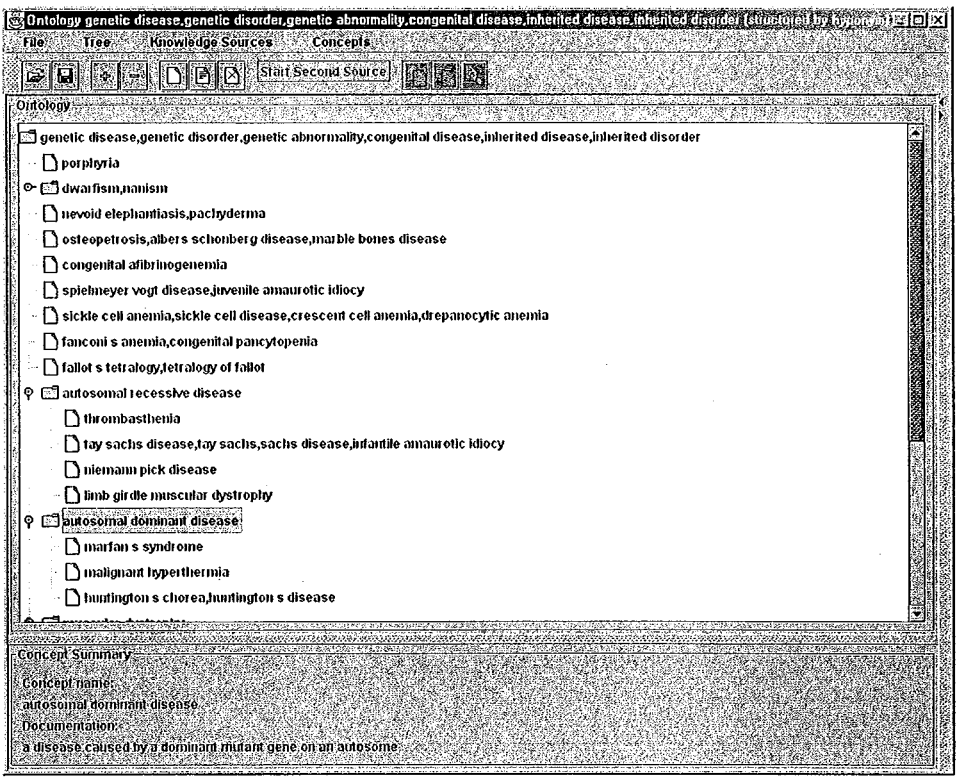


FIGURE 9. A hyponymy hierarchy rooted in the concept "genetic disease" is produced.

disease", the knowledge engineer starts to add simple slots describing symptoms, general prevalence information and the genetic model of the disease (Figure 10).

This scenario has shown IMPS being used to experiment with different representations of domain knowledge. XML DTDs were used as sources of domain-specific ontological knowledge. In this role, they proved to be rich sources of concept descriptions and attributes, however, they were weak in terms of specifying consistent formal relationships between concepts. IMPS was used to detect and correct this feature as concepts from DTDs were added into an ontology. It was also used to rapidly explore the features of domain knowledge in one portion of the proposed ontology that constrain the knowledge representation in another, before any firm commitment was made.

6. Related work

Tools with explicit support for ontology merging are being developed (mostly in the last one or two years), and these tools are becoming increasingly sophisticated. These tools seem to stem from a similar awareness of the issues of semantic integration and the potential for reuse of existing knowledge structures.

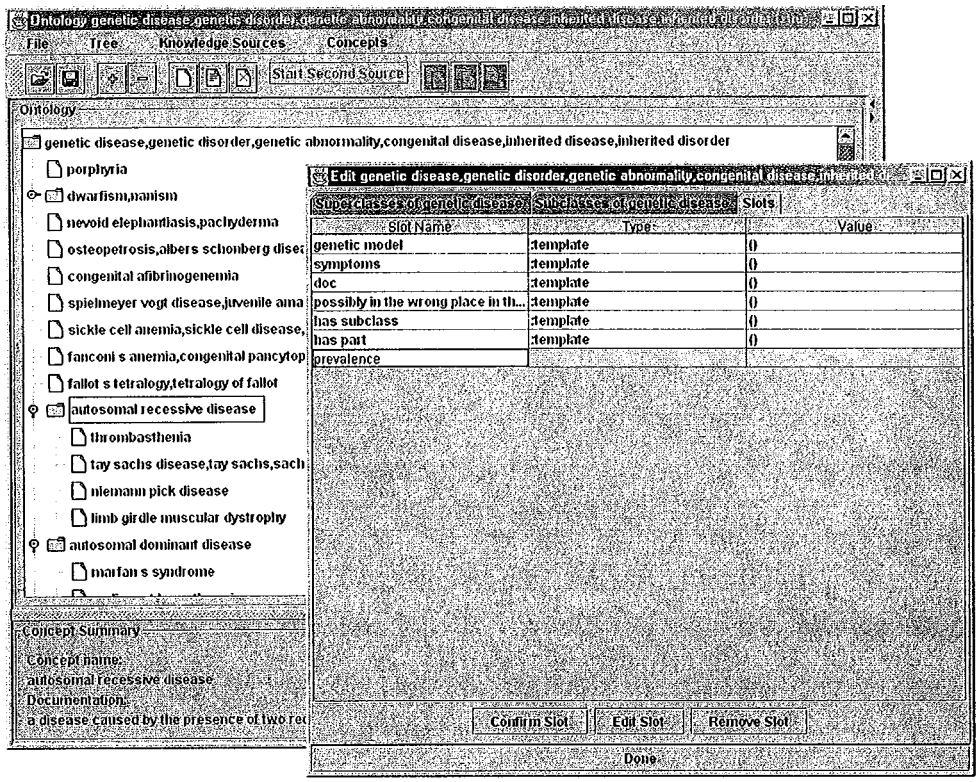


FIGURE 10. Slots for symptoms, prevalence and genetic model are added to the concept "genetic disease". These will be inherited by the subclasses of the concept.

Recently, Fridman Noy and Musen (1999) presented SMART, a semi-automatic tool which aims to support ontology merging and alignment. The knowledge model underlying the system, like the IMPS knowledge model, is compatible with the OKBC protocol (Chaudri *et al*., 1998). SMART is envisioned as becoming part of Protégé-2000, the latest implementation of the Protégé knowledge modelling environment. SMART merges existing AI ontologies represented in frames in Protégé-2000. Both IMPS and SMART supplement the OKBC knowledge model by adding another layer to it to support primitive merging operations so that these operations can be performed in one step. At present there is only one merging operation that a user can choose to perform using SMART—the "shallow copy" operation that copies a frame from the source ontology to the target ontology without copying any of the frames that that frame refers to. However, SMART also performs some merging actions automatically, makes suggestions to guide the user, checks for conflicts and proposes solutions to them.

DODDLE (Yamaguchi, 1999) is a rapid development environment for constructing domain ontologies. Like IMPS and SMART, DODDLE focuses on the construction of a hierarchically structured set of domain concepts. Unlike the other systems, DODDLE

does not represent concepts with definitions or any other internal structure—they are simply keywords. IMPS and DODDLE both use WordNet as a machine-readable dictionary to create an initial model of a domain, both also recognizing that on its own this is not sufficient input to create an ontology. DODDLE requires the user to supply much more domain knowledge—in the form of single terms and also concept trees.

Chimæra (McGuinness *et al*., 2000) is a tool for large-scale ontology merging, diagnosis and maintenance that has been developed as part of DARPA's high performance knowledge bases (HPKB) program. Like IMPS, Chimæra attempts to create class taxonomies from existing web sources, using Yahoo! Shopping, Lycos, Topica, Amazon and UN/SPSC as mines of taxonomic information. However, the emphasis in this mining effort is not on reasoning, as in IMPS, but on the use of the knowledge to support web site browsing, organization and search. Chimæra offers extensive diagnostic support for ontology building and support for creating and editing disjoint partition information.

There is a degree of similarity to be found in the emerging ontology tools; the approaches focus firstly on class merging with some consideration of slots and the relationships between classes, principally inheritance relationships. The construction of taxonomic inheritance hierarchies seems to be a common activity, with IMPS somewhat diverging from the approach by providing additional support for partonomy hierarchies. This can be attributed to the unique task-based and inference-driven approach that IMPS takes, using task models to motivate and direct the extraction of concepts and relationships to support specific types of problem-solving inference.

In a recently developing field, the use of web sources with ontological qualities appears to be the most recent development! Both IMPS and Chimæra attempt to extract such information and raise the level of representation. However, IMPS uses richer and more diverse ontological sources, extracting more knowledge from them—classes, relationships between them and their slots and values in some cases.

This is not the only set of tools that IMPS can be compared with. As well as being an ontology merging tool, IMPS is also a multi-agent implementation making use of networked resources. In Crow (2000), we situate IMPS with respect to other agent architectures for information retrieval and fusion [looking specifically at InfoSleuth (Nodine, Fowler & Perry, 1998), RETSINA (Sycara, 1999), SIMS (Knoblock & Ambite, 1997), Ariadne (Knoblock *et al*., 1998) and KRAFT (Gray *et al*., 1997)]. Although the aim of these systems may be slightly different from that of IMPS, we hope to find areas of consensus in the architectures and interaction mechanisms that are successful in supporting a web-based information agent architecture, and also to examine more sophisticated agent systems to guide the future development of the IMPS agent model.

7. Future directions

Future directions for the IMPS architecture will be driven by the vision of a system performing principled knowledge-based inference for a variety of tasks using knowledge extracted from the web. Developments that would facilitate this include general refinement of the ontology representation to include features such as facets, individuals, axioms and partitions. We are also interested in extracting and representing a wider

range of relationships between concepts to support inference. However, this presents the challenge that while hyponymy, the "is-a" relationship is an agreed ontological primitive (see, for example, Duineveld, Stoter, Weiden, Kenepa & Benjamins, 2000), the semantics of other relationships are less clear cut and subject to different interpretations. The existence of a relatively stable set of problem-solving inferences which include but are not limited to classificatory inferences indicates that human problem-solving knowledge has a wider scope. Concepts like causality are central to our understanding of the world but seem to be under-represented in general knowledge structures.

The widespread use of rich semantic markups may change this situation considerably, establishing a canon of useful semantic relationships. With these developments in mind, IMPS has been designed as an open-ended architecture to allow the plug-and-play addition of code modules to extract knowledge from new types of source. As we noted, the proposed semantic markup schemes DAML and OIL are based in XML—techniques to parse these kinds of documents could be based in the existing XML capability in the system. Additionally, the use of the OKBC knowledge model opens up the possibility of merging ontologies in any OKBC compliant representation system on the same machine or over the network (McGuinness *et al*., 2000). This means that IMPS has the potential to access through OKBC ontologies written using Cyc, Protégé, LOOM, Theo, SIPE-2, Ocelot, Ontolingua, ATP (a theorem prover), file systems, Tuple-KB and CLOS (Chaudhri, Thomere & Grosso, 1998).

In the IMPS prototype, we currently focus on the components of a domain knowledge schema. As well as trying to perform a general match between the whole schema and the ontological knowledge available, providing useful concepts and relationships, we recognize that the domain knowledge required to support most problem-solving methods is heterogeneous. Therefore, we use the requirements associated with a specific role to extract domain knowledge. This creates parts of a heterogeneous purpose-built knowledge base. The next logical step is to support the configuration of these parts into a whole knowledge base. If we developed the system so that it accepted several different keywords from the user for different roles, a complete ontology could be constructed by mining each role in the appropriate directions and configuring them together.

We have discussed the use of the knowledge in IMPS for inferencing, but in order to achieve that we will have to populate the ontological structures created. Ontologies typically do not contain factual information about the domain, merely making assertions about the kinds of objects and relationships that are to be found there. A particularly exciting area with respect to fleshing out the knowledge base is the use of documents with semantic markup. We have described the relationship between XML documents and the DTDs that define them. It could be said that the relationship between an XML DTD and an XML document can be mapped directly onto the relationship between an ontology and the knowledge base it defines. Therefore, instantiating the portions of an ontology that are directly based on a DTD with domain knowledge is a simple matter of parsing the relevant XML documents and adding them into the knowledge base. What would present more of a challenge is the situation where the user constructing the ontology has used parts of the DTD (or indeed any ontological knowledge source) and modified those parts. In this situation, the system must keep a track of the operations that have been performed to produce the current ontological entities from the original document. Those

same operations must be performed on the XML entities that are to fill the knowledge base. A promising approach for accomplishing this can be seen in the information integration literature. Systems such as SIMS (Ambite *et al*., 1998), Ariadne (Knoblock *et al*., 1998), InfoSleuth (Nodine *et al*., 1998) and RETSINA (Sycara, 1999) maintain global data models or ontologies which are used to integrate the individual data models of several heterogeneous information sources using ontology mappings. This approach could be modified so that this ontological model stores complex transformations required to transform information from the original source to integrate it into the global data model (or in this case ontology).

The extension of the IMPS system to extract ontological knowledge for a fuller range of task is dependent, to a large extent, on the success of attempts to extract different semantic relations between concepts, and to compose ontologies in a more sophisticated way from smaller heterogeneous ontological structures. A second goal with relation to the knowledge intensive tasks represented in IMPS is a more principled and sophisticated expression of those tasks. In order to accomplish this, we could use a unified knowledge component representation language such as unified problem-solving method description language (UPML) (Fensel *et al*., 1999*a*; Fensel, Benjamins, Motta & Wielinga, 1999*b*; Benjamins, Wielinga, Wielemaker & Fensel, 1999). The use of a language such as UPML to describe tasks in IMPS could allow the formalization of the existing task descriptions and the use of new task descriptions from other sources.

Additionally, the proposed extension of IMPS to include agents performing inferences will require the mapping from task model requirements to the component inferences of problem-solving methods. UPML is an architectural description language for describing specific kinds of reasoning components. There are four component types in the language—tasks, PSMs, domain models and ontologies. Its power stems from the fact that it allows the integrated description of these diverse knowledge components. With the existence of repositories of problem-solving methods, as described by Benjamins *et al*. (1999), comes the possibility of importing and configuring PSM fragments or full PSMs for applications from external sources. Benjamins *et al*. (1999) describe how UPML can enable the matching of PSM components to task requirements and the configuration of a problem solver from heterogeneous and distributed components.

This paper has presented a multi-agent architecture for the construction of task-oriented domain ontologies for problem solving using web knowledge sources. This work draws upon several fields of research—knowledge engineering, the study of ontologies, software agency and other approaches to the challenges raised by the web. The IMPS architecture has been developed with the aim of exploring some of the increasing areas of overlap between the concerns and strengths of these disciplines. We believe that the potential for the fusion of ideas into powerful and useful technologies is rich enough to guarantee a continued interest in the synthesis of issues presented here. There is no plausible vision of the future in which agents, ontologies and the web will not play a role.

This research was carried out as part of Louise Crow's Ph.D., which was supported by a University of Nottingham Research Scholarship.

References

ARPIREZ, J., GOMEZ-PEREZ., LOZANO, A. & PINTO, S. (1998). (Onto)2Agent: an ontology-based WWW broker to select ontologies. In A. GÓMEZ-PÉREZ & V. R. BENJAMINS, Eds. *Proceedings of the 13th Biennial European Conference on Artificial Intelligence* (*ECAI98*) *Workshop on Applications of Ontologies and Problem Solving Methods*. pp. 16–24. Wiley. Brighton, England.

BATEMAN, J., MAGNINE, B. & RINALDI, F. (1994). The generalised Italian, German, English upper model. In *Proceedings of The Eleventh European Conference on Artificial Intelligence (ECAI'94) Workshop on Comparison of Implemented Ontologies*, Amsterdam.

BENJAMINS, V. R. & FENSEL, D. (1998). Community is knowledge! in $(KA)^2$. In B. GAINES & M. MUSEN, Eds. *Proceedings of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop* (*KAW'98*), Banff, Canada. URL: http://ksi.cpsc.ucalgary.ca/KAW/KAW98/benjamins1/.

BENJAMINS, V. R., WIELINGA, B., WIELEMAKER, J. & FENSEL, D. (1999). Brokering problem-solving knowledge on the internet. In D. FENSEL *et al*., Eds. *Proceedings of the European Knowledge Acquisition Workshop* (*EKAW'99*), Lecture Notes in Artificial Intelligence (LNAI), Vol. 1621. Berlin: Springer. URL: ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/ekaw99.benjamins.ps

BERNERS-LEE, T. (1999). Challenges of the second decade. WWW8. Toronto.

BERNSTEIN, F., KOETZLE, T., WILLIAMS, G., MAYER, E., BRYCE, M., RODGERS, J., KENNARD, O., SIMANOUCHI, T. & TASUMI, M. (1977). The protein data bank. *Journal of Molecular Biology*, **112,** 535–542.

BRAY, T., PAOLI, J. & SPERBERG-MCQUEEN, C. M. (1998). Extensible Markup Language (XML) 1.0 W3C Recommendation REC. xml-19980210 URL: http://www.w3.org/TR/1998/REC-xml-19980210.

BYLANDER, T. & CHANDRASEKARAN, B. (1998). Generic tasks in knowledge based reasoning: the right level of abstraction for knowledge acquisition. In B. GAINES & J. BOOSE, Eds. *Knowledge Acquisition for Knowledge Based Systems*, Vol. 1, pp. 65–77. London: Academic Press.

CHASE, W. & SIMON, H. (1973). The Mind's Eye in Chess. In W. G. CHASE, Ed. *Visual Information Processing*, pp. 251–258. New York, Academic Press.

CHAUDHRI, V., THOMERE, J. & GROSSO, W. (1998). Open knowledge base connectivity specification FAQ (frequently asked questions). URL: http://www.ai.sri.com/ = okbc/okbc-faq/.

CHAUDHRI V., FARQUHAR, A., FIKES, R., KARP, P. & RICE, J. (1998). *Open knowledge base connectivity 2.0.3. specification document*. Technical Report KSL-98-06, Knowledge Systems Laboratory, Stanford University.

CHI, M. T. H., FELTOVICH, P. J. & GLASER, R. (1981). Categorisation and representation of physics problems by experts and novices. *Cognitive Science*, **5,** 121–152.

CROW, L. R. (2000). *Software agents for internet-based knowledge engineering*. Ph.D. Thesis, School of Psychology, University of Nottingham.

CROW, L. R. & SHADBOLT, N. R. (1998). IMPS—Internet agents for knowledge engineering. In B. GAINES & M. MUSEN, Eds. *Proceedings of the 11th Knowledge Acquisition for Knowledge Based Systems Workshop* (*KAW'99*). Calgary: SRDG Publications. Available at: http://ksi.cpsc.ucalgary.cal/KAW/KAW98/crow/.

CROW, L. R. & SHADBOLT, N. R. (1999). Acquiring and structuring web content with knowledge level models. In *The Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling, and Management* (*EKAW 98*). Lecture Notes in Artificial Intelligence, Vol. 1621, pp. 85–102. Berlin: Springer-Verlag. Available at: http://www.aifb.uni-karlsruhe.de/ekaw99/LNCS.html.

CROW, L. R. & SHADBOLT, N. R. (2000). Exploiting the ontological qualities of web resources: task-driven agents structure knowledge for problem solving. *Proceedings of the 4th International Workshop on Cooperative Information Agents* (*CIA2000*). Lecture Note in Artificial Intelligence, Vol. 1860. Berlin: Springer.

CUPIT, J. & SHADBOLT, N. R. (1994). Representational redescription within knowledge intensive data-mining. In R. MIZOGUSHI *et al*., Eds. *Proceedings of the 3rd Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, JKAW'94*.

DAHLGREN, K. (1995). A linguistic ontology. *International Journal of Human-Computer Studies*, **43,** 809–818.

DUINEVELD, A., STOTER, R., WEIDEN, M., KENEPA, B. & BENJAMINS, V. R. (2000). Wondertools? A comparative study of ontological engineering tools. *International Journal of Human-Computer Studies*, **52**, 1111–1133.

FARQUHAR, A., FIKES, R. & RICE, J. (1996). The ontolingua server: a tool for collaborative ontology construction. In B. GAINES & M. MUSEN, Eds. *Proceedings of The 10th Knowledge Acquisition for Knowledge-Based Systems Workshop* (*KAW'96*). Banff, Canada: SRDG Publications. URL: http://ksi.cpsc.ucalgary.ca/KAW/KAW96/farquhar/farquhar.html.

FENSEL, D. (2000). Relating ontology languages and web standards. In J. EBERT *et al*., Eds. *Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik*, Modellierung 2000, St. Goar, April 5–7, Foelbach Verlag, Koblenz.

FENSEL, D., BENJAMINS, V. R., DECKER, S., GASPARI, M., GROENBOOM, R., GROSSO, M., MUSEN, M., MOTTA, E., PLAZA, E., SCHREIBER, G., STUDER, R. & WIELINGA, B. (1999). The component model of UPML in a nutshell. *WWW Proceedings of the 1st Working IFIP Conference on Software Architectures* (*WICSA1*), San Antonio, TX, USA. URL: ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/upml.ijfip.ps.

FENSEL, D., BENJAMINS, V. R., MOTTA, E. & WIELINGA, B. (1999*b*). UPML: a framework for knowledge system reuse. In *Proceedings of the International Joint Conference on AI* (*IJCAI-99*). URL: ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/upml.ijcai.ps.

FENSEL, D., HORROCKS, I., VAN HARMELEN, F., DECKER, S., ERDMANN, M. & KLEIN, M. (2000). OIL in a nutshell. URL: http://www.ontoknowledge.org/oil/papers.shtml#1.

FENYO, D. (1999). BioML—The biopolymer markup language. *Bioinformatics*, **15,** 339–340.

FININ, T., LABROU, Y. & MAYFIELD, J. (1997). KQML as an agent communication language. In J. BRADSHAW, Ed. *Software Agents*, pp. 291–316. Cambridge, MA: MIT Press.

FRIDMAN NOY, N. & HAFNER, C. (1997). The state of the art in ontology design: a survey and comparative review. *AI Magazine*, **18,** 53–74.

FRIEDMAN-HILL, E. J. (1998). *Jess, the Java expert system shell*. Technical Report, SAND98-8206 Sandia National Laboratories, Livermore. http://herzberg.ca.sandia.gov/jess.

FROST, H. R. (1996). *Documentation for the Java*(*tm*) *agent template*, Version 0.3. Stanford University. URL: http://edr.stanford.edu/ABE/documentation/index.html.

GANGEMI, A., STEVE, G. & GIACOMELLI, F. (1996). ONIONS: an ontological methodology for taxonomic knowledge integration. In VET P. E. VAN DER Ed. *Proceedings of the Workshop on Ontological Engineering European Conference on Artificial Intelligence* (*ECAI96*).

GENESERETH, M. & FIKES, R. (1992). *Knowledge interchange format version* 3.0 *reference manual*. Technical Report, Logic-92-1. Computer Science Department, Stanford University.

GOBET, F. & SIMON, H. A. (1995). *Chunks in chess positions*: *recall of random and distorted positions*. Complex Information Paper #518, Carnegie Mellon University, Pittsburgh.

GOBET, F. & SIMON, H. A. (1996). The roles of recognition processes and look-ahead search in time-constrained expert problem solving: evidence from grand-master-level chess. *Psychological Science*, **7,** 52–55.

GRAY, P., PREECE, A., FIDDIAN, N., GRAY, W., BENCH-CAPON, T., SHAVE, M., AZARMI, N., WIEGAND, M., ASHWELL, M., BEER, M., CUI, Z., DIAZ, B., EMBURY, S., HUI, K., JONES, D., KEMP, G., LAWSON, E., LUNN, K., MARTI, P., SHAO, J. & VISSER, P. (1997). KRAFT: knowledge fusion from distributed databases and knowledge-bases. *Proceedings of the 8th International Workshop on Database and Expert Systems Applications* (*DEXA'97*), pp. 682–691.

GUARINO, N. (1996). Understanding, Building and Using Ontologies. In B. GAINES and M. MUSEN (Eds.) *Proceedings of The Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96)*, pp. 29-1–29-11. Banff, Canada: SRDG Publications. URL: http://ksi.cpsc.ucalgary.ca/KAW/KAW96/guarin

HENDLER, J. (2000) DAML Description. URL: http://dtsn.darpa.mil/iso/programtemp.asp?mode = 34.

JOHNSON, P. E., DURAN, A. S., HASSEBROCK, F., MOLLER, J., PRIETULA, M., FELTOVICH, P. J. & SWANSON, D. B. (1981). Expertise and error in diagnostic reasoning. *Cognitive Science*, **5,** 235–283.

KNOBLOCK, C. & AMBITE, J. (1997). Agents for information gathering. In J. BRADSHAW (Ed.) *Software Agents*. AAAI/MIT Press. Menlo Park, CA, pp. 347–374

KNOBLOCK, C., MINTON, S., AMBITE, J. E., ASHISH, N., MODI, P., MUSLEA, I., PHILPOT, A. & TEJADA, S. (1998). Modellling web sources for information integration. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI. pp. 211–218.

KORN, F. (1996). A taxonomy of browsing methods: approaches to the lost in concept space problem. Department of Computer Science M.Sc. Dissertation. University of Maryland.

LAMPING, L., RAO, R. & PIROLLI, P. (1995). A focus + context technique based on hyperbolic geometry for visualising large hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*.

LARESGOITI, I., ANJEWIERDEN, A., BERNARAS, A., CORERA, J., SCHREIBER, A. Th. & WIELINGA, B. (1996). Ontologies as vehicles for reuse: a mini-experiment. In B. GAINES & M. MUSEN, Eds. *Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96)*. Banff, Canada: SRDG Publications. URL: http://ksi.cpsc.ucalgary.ca/KAW/KAW96/laresgoiti/k.html.

LENAT, D. (1995). Cyc a large-scale investment in knowledge infrastructure. *Communications of the ACM (CACM)*, **33**(8), 30–49.

LUKE, S., SPECTOR, L., RAGER, D. & HENDLER, J. (1997). Ontology-based web agents. *Proceedings of the 1st International Conference on Autonomous Agents*, pp. 59–66.

MAYUYAMA, H., TAMURA, K. & URAMOTO, N. (1999). *XML and Java*: *Developing Web Applications*. Reading, MA: Addison-Wesley.

MCGUINNESS, D., FIKES, R., RICE, J. & WILDER, S. (2000). An environment for merging and testing large ontologies. In A. G. COHN, F. GUNCHIGLIA & B. SELMAN, Eds. *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*. Breckenridge, CO, USA. 12–15 April. URL: http://www.kr.org/kr/kr00/

MILLER, G., BECKWITH, R., FELLBAUM, C., GROSS, D. & MILLER, K. (1993). Introduction to WordNet: An On-line Lexical Database. URL: http//www.cogsci.princeton.edu/ ~wn/

NEWELL, A. (1982). The knowledge level. *Artificial Intelligence*, **18,** 87–127.

NEWELL, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

NODINE, M., PERRY, B. & UNRUH, A. (1998). Experience with the infosleuth agent architecture. In *Proceedings of the 1998 Conference of the American Association for Artificial Intelligence* (*AAAI-98*) *Workshop on Software Tools for Developing Agents*, pp. 63–72. Menlo Park, CA: AAAI Press.

O'LEARY, D. E. (1997). Impediments in the use of explicit ontologies for KBS development. *International Journal of Human-Computer Studies*, **46,** 327–337.

PETERSON, G. (1999). WordNet 1.6 Vocabulary Helper. URL: http://www.notredame.ac.jp/cgi-bin/wn.

PISANELLI, D., GANGEMI, A. & STEVE, G. (1999). A medical ontology library that integrates the UMLS metathesaurus. In *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, pp. 239–248.

RAPOZA, J. (2000). DAML could take search to a new level. *PC Week Online*, **17,** 33.

RUSS, T., VALENTE, A., MACGREGOR, R. & SWARTOUT, W. (1999). Practical experiences in trading off ontology usability and reusability. In *Proceedings of the 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'99)*. Banff, Canada. URL: http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/Russ1/kaw99-4-14.pdf.

SCHREIBER, G., AKKERMANS, H., ANJEWIERDEN, A., DE HOOG, R., SHADBOLT, N., VAN DE VELDE, W. & WIELINGA, B. (2000). *Knowledge Engineering and Management*: *The CommonKADS Methodology*. Cambridge, MA: MIT Press.

SCHUYLER, P. HOLE, W., TUTTLE, M. & SHERETZ, D. (1993). The UMLS metathesaurus: representing different views of biomedical concepts. *Bulletin Medical Library Association*, **81,** 217–222.

SHAW, M. & GAINES, B. (1989). Comparing conceptual structures: consensus, conflict, correspondence and contrast. *Knowledge Acquisition*, **1,** 341–363.

SYCARA, K. (1999). In context information management through adaptive collaboration of intelligent agents. In M. KLUSCH, Ed. *Intelligent Information Agents*: *Agent-based Information Discovery and Management on the Internet*, pp. 78–99. Berlin, Springer.

USCHOLD, M., CLARK, P. HEALY, M., WILLIAMSON, K. & WOODS, S. (1998*a*). An experiment in ontology reuse. In B. GAINES & M. MUSEN, Eds. *Proceedings of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'98),* Banff, Canada. URL: http://ksi.cpsc.ucalgary.ca/KAW/KAW98/uschold/.

USCHOLD, M. & JASPER, R. (1999). A framework for understanding and classifying ontology applications. In V. R. BENJAMINS, B. CHANDRASEKARAN, A. GÓMEZ-PÉREZ, N. GUARINO & M. USCHOLD Eds. *Proceedings of The International Joint Conference on Artificial Intelligence (IJCAI-99) Workshop on Ontologies and Problem-Solving Methods (KRR5)*, Stockholm, Sweden. URL: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/.

USCHOLD, M., KING, M., MORALEE, S. & ZORGIOS, Y. (1998*b*). The enterprise ontology. *Knowledge Engineering Review*, **13,** 31–89.

VAN HARMELEN, F. & FENSEL, D. (1999). Practical knowledge representation for the web. In *Proceedings of The International Joint Conference on Artificial Intelligence (IJCAI'99) Workshop on Intelligent Information Integration.* Proceedings at http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-23/

VAN HEIJST, G., SCHREIBER, A. Th. & WIELINGA, B. (1997). Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, **46,** 183–292.

Visual Genomics (1999). Bioinformatic Sequence Markup Language (BSML) and the BSML Browser & Tools: XML Applications for Management, Communication, and Interactive Visualisation of Genomic Data. URL: http://www.visualgenomics.com/bsml/index.html.

WEINSTEIN, P. & BIRMINGHAM, W. (1999). Comparing concepts of differentiated ontologies. In *Proceedings of the 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'99)*, Banff, Canada. URL: http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/Bermingham1/kaw99_final.pdf.

WIEDERHOLD, G. & JANNINK, J. (1999). Composing diverse ontologies. *IFIP Working Group on Databases*, *8th Working Conference on Database Semantics (DS-8)*, Rotorua, New Zealand.

Paper accepted for publication by Editor, Prof. B. Gaines.