Draft
Written October 2004
Last Revision October 2006

ONTOLOGICAL MODELLING: STATE OF THE ART, UNRESOLVED ISSUES AND NEW RESEARCH DIRECTIONS

Asoka S. Karunananda[1] and George Rzevski[2]

[1] Faculty of Information Technology, University of Moratuwa, Sri Lanka
[2]The Open University, Milton Keynes and Magenta Corporation, London

Abstract

The paper contains a review of current research into ontological modelling, as reported in Semantic Web, Multi-Agent Systems and Web Services literature, conducted with a view to identifying unresolved issues. The key findings are that current ontology design methods do not take into account (a) the variety of perceptions of the problem domains held by different problem owners and (b) the evolution of knowledge with time. Authors suggest how these issues could be resolved with the help from insights gained by studying philosophical foundations of ontology. The paper concludes with suggestions for further research into ontological modelling.

Key Words Ontology, Agents, Intelligent Systems, Mental Attributes, Knowledge Evolution, Semantic Web

1 INTRODUCTION

The question how to describe the world in which we live and work (the fundamental problem of ontology) puzzled philosophers from the very early days of the development of human knowledge. The great minds spent their lifetimes attempting to articulate answers to this question, building in the process a formidable edifice of human knowledge.

As we began to model the world around us by computer programs, we ignored the experience accumulated through centuries of ontological studies and decided to describe complexities of reality in terms of rigid mathematical algorithms translated into computer readable procedures. Until late 1970s, applied computing was considered almost exclusively in these terms although there was always a minority of computer scientists interested in Knowledge-Based Systems and Artificial Intelligence seeking methods for dealing with uncertainty. For example, more than twenty years ago Newell (1982) suggested that the key unresolved issues in computing are at the *knowledge level*. Much effort has gone since then into researching how to solicit, store, process and utilise knowledge. Although we have progressed from procedural to functional and declarative languages; from databases, via rule-based knowledge bases to the concept of ontology, the state of the software *practice* has not significantly changed. Software industry still produces masses of rigid code.

The thesis of this paper is that philosophical foundations of ontology provide a plethora of ideas for improving computer readable models of complex problem domains. The implementation of these ideas may lead to improved, human-centred ontological models, as outlined in closing sections of this paper.

The paper outlines fundamental concepts of ontology, critically reviews ontology-related literature, identifies main unresolved issues in ontological modelling and suggests directions for further research. The paper is organised as follows.

Section 2 discusses fundamentals of concepts of ontology. Section 3 is a critical review of ontology literature, which covers ontology-based systems in Semantic Web, Multi-Agent Systems and Web Services research, followed by a discussion on ontological tools in general. Each section is provided with a summary table, highlighting the underlying basis and technology of ontology development tools/environments. Section 5 identifies the issues in ontological modelling and Section 6 contains proposals how these issues could be addressed. Section 7 concludes the paper with a note on further work.

2 FUNDAMENTAL CONCEPTS

Ontology is a *conceptual* description of the world in which we live and work. In general, ontology deals with *concepts* such as being, existence, enterprise, operator, resource, demand, machine, order and the like, as well as with *relations* between these concepts. In general, ontology does not concern itself with specific instances of concepts or values of attributes.

In defining ontology, computer scientists use the term *classes* instead of *concepts*. In a computing parlance ontology consists of classes of *objects*, classes of *relations* between objects, classes of their *attributes* and, in some cases, classes of their *behaviours*, (which may be given in terms of rules or scripts). Instances of these classes are normally not considered although some authors label instances as classes following convention accepted in object-oriented programming. V*alues* of attributes of objects are always kept outside ontology. The separation of conceptual knowledge from the details (facts, values) is a considerable advance in respect to the earlier paradigm of rule-based knowledge bases; it reduces the effort of constructing ontology.

However, there is still no general agreement among computer sciences on the definition of ontology. The frequently used definition provided by Gruber (1993) which states that ontology is “an explicit specification of a conceptualization”, appears to some authors to be somewhat obscure and Guarino (1997) pointed out that at the time of writing his paper at least seven definitions for the term ontology were in circulation. Two years later Nwana & Ndumu (1999) identified the ontological problem as one of the key issues in Multi-Agent Systems and yet Hendler (2001) maintained that in 2001, in other words, eight years after Gruber’s definition, ontology was still one of the ill-defined terms in computer science.

In addition to Semantic Network approach to organising ontology, computer scientists have also taken other approaches such as Frames (OIL: Horrocks *et al*., 2000), First Order Logic and declarative programming (OWL: McGuinness et al. 2004).

In philosophical terms, Aristotle defines ontology as the *science of being or existence with regard to the aspects of being or existence* (http://en.wikipedia.org/wiki/Ontology; Hendler, 2001). Whilst Aristotle has focused on the physical phenomenon of existence, or being, Buddhist philosophical view, articulated as early as 563 BC (Sowa 1995; Silverman 2002), has been that existence is a collection of *mental* and physical phenomena (McEvilley, 2002; Sitagu 2004; Gorkom 2004). This particular finding points to possible answers to key issues in ontological modelling which are currently unresolved, as discussed in Section 5 of this paper.

Perhaps even more importantly, Buddhist doctrine maintains that each individual may have a somewhat different perception of the world (which could be justified, to use current terminology, by the differences in sensor sensitivity, emotional makeup and cognitive capability of individuals) and therefore it is problematic to talk about an objective ontology of a complex problem domain. Anyone attempting to solicit contributions to and validate results of building ontology in practical situations must have experienced this phenomenon. As a rule, problem-domain stakeholders exhibit surprising differences in understanding various aspects of the problem they own (Rzevski 2005).

Therefore, in problem domains where a consensus on the nature of the problem cannot be achieved among stakeholders, there is a strong case for building a number of ontologies each reflecting a viewpoint of a coherent group of stakeholders (e.g., economists, engineers, marketing experts, employees) and then finding a way of merging these “partial” ontologies into an overall problem domain ontology.

Early Buddhist philosophical works also emphasised the importance of knowledge development and growth, that is, the evolution of knowledge. The notion of continuously evolving knowledge is particularly important at present when we experience a number of paradigm shifts related to the transition to Information Economy (Rzevski 2003).

In the recent past ontological modelling has been a major research theme in several areas of computer science.

Certain researchers hold the view that in philosophical terms, ontology is synonymous with the real would, yet in computing ontology is only a model and therefore the two should be differentiated. This view is unacceptable to authors of this paper for the simple reason that any description of the real world is by definition a model, and only a model. It is not possible to think about or articulate the reality without building its model. The same applies to philosophers and computer scientists, or anybody else.

Models, of course, differ depending on their purpose and on the knowledge and skills of model builders. If one is allowed some generalisation, philosophers tend to produce all embracing models that lack in detail whilst computer scientists are inclined to ignore computationally inconvenient aspects of reality, such as mental attributes, relativity of knowledge and its evolution. The difference is in inclusions and omissions rather than in principles.

There are however serious disputes among philosophers as well as among computer scientists on what should and what should not be part of ontology. For example, according to Quine (1961) ontological commitment of logical theory is an account of what exists, yet Guarino (1997) has clearly argued that in building ontology, we should go beyond what exits and capture the *structure* of what exists.

The position of authors on the issue of what is and what is not ontology is as follows.

Ontology is a conceptual description (a conceptual model) of the real world which reflects

- The purpose of ontology
- The perception of the real world of the ontology builder at the time of ontology building

Ontology emphasises concepts, such as objects, their attributes (both mental and physical), their behaviour and their relations, ignoring specifics, such as instances of objects, values of their attributes and relations, and algorithms that implement their behaviour.

3 A REVIEW OF LITERATURE

Early beginnings (the 1980s) were characterised by the construction of ontologies for very specific problem domains such as geography or natural language processing. The real explosion of interest in ontological research occurred only when the work on the development of Semantic Web took off in the late 1990s. The key idea of Semantic Web, ie that computers will be interpreting the meaning of data rather than just following a trail of addresses, has naturally a broad appeal; and since ontology has to play an important role in the realisation of this idea, it is not surprising that ontological modelling emerged as a priority research topic. In the discipline of Multi-Agent Systems (MAS) ontology also plays an increasingly important role as a formalised repository of conceptual knowledge which can be consulted by agents before making decisions. The discipline of Web Services, which began life as an independent field of research, is now heavily interlinked with ontology-based MAS. Two areas of ontological research, Ontology Translation and Ontology Development Tools appear to grow exceptionally fast.

3.1 Ontology-Based Systems for Specific Problem Domains

In the early days of development the computational ontological systems were very much domain specific. These systems were developed well before the emergence of general environments for ontological modelling, and as a result they have been programmed from the scratch, using languages such as Prolog or Lisp.

One of the first and simplest early ontology-based systems was Chat-80, a question-answering system, which was designed with ontology of the micro world of geography (Warren & Pereira 1982). This was one of the first major Prolog-based systems to demonstrate natural language processing with a geographical ontology. It defined ontology with the topmost category *geographical features*, which had three sub categories *area*, *point* and *line*. There were five sub categories identified for *area*, and six sub categories for *point*. *Line* had two sub categories, each with further partitions. With just 20 categories this Ontology was small and very specific. Chat-80 is perhaps too simple to be considered as a good example of ontology.

In contrast, Cyc (Lenat, 1995) has quite a comprehensive ontology based on first-order predicate logic. The name Cyc is an acronym derived from the world encyclopaedia. In the early days, the most general category in Cyc was *thing*, which had three sub categories, namely, *individual objects, intangibles*, and *represented things*, followed by further sub categories. However, in recent publications Cyc knowledge-base was describes as comprising *things* as the top level category and *intangible* and *individual* things as the immediate subcategories. See, for example, the website: http://www.cyc.com/cyc/technology/whatiscyc_dir/whatdoescycknow.

.

Cyc ontology is not a hierarchy, but an acyclic graph, showing many clusters of categories. These categories cover physical categories and some mental categories such as emotion, perception and beliefs. With reference to those categories Cyc has defined over one million commonsense axioms. The objective of the Cyc project is to assemble a comprehensive ontology for enabling a variety of knowledge-intensive products and services to work together. Lenat states that Cyc can be considered as a huge expert system for a domain that spans all everyday objects and actions. Cyc expert system architecture includes eight components, and among these the Knowledge Base (KB) is the repository of knowledge, which refers to Cyc ontology (Siegel et at. 2004). Cyc KB approximately has 2.5 million facts and rules relating to more than 155,000 concepts. The Cyc KB ontology includes subject specific knowledge (e.g. biology, military, chemistry, etc.) and more general commonsense knowledge (e.g. buying, selling, kinship relations, etc.), as well as lexical and grammatical knowledge required to enable natural language processing capabilities of Cyc.

Free version of Cyc knowledge base is available as OpenCyc. Sowa (1995) has however criticised Cyc Ontology for its incompleteness and lack of philosophical justification for identification of categories. This issue has been recognized by Cycorp, Inc. itself (Reed & Lenat, 2002) and no satisfactory answer is given. We agree with Sowa that lack of philosophical justification is an issue common to most ontological systems.

WordNet is a lexical database available online for natural language processing pertaining to English language (Miller, 1995). It comprises ontology of most general concepts in English language; *nouns, verbs, adjectives* and *adverb*s together with set of synonyms related to each. The WordNet defines the vocabulary of English language as a set of pairs comprising word *form* and *sense* and it comprises more than 166,000 word form-sense pairs. Ontology of WordNet also includes semantic relationships such as *synonym, antonym, hyponym, meronym, holonym* and *troproponym*. In this sense, WordNet offers just six types of semantic relations, whereas Cyc has thousands (Lenat *et al.,* 1995). Further, as compared with Cyc, WordNet does not have much detail about categories or terms in its ontology. Nevertheless, WordNet provides the most widely used ontology for natural language processing in the world. Since ontology-based natural language processing is likely to be a key factor in application areas such as e-commerce and semantic web, WordNet is an important initiative. Furthermore, WordNet has inspired the development of various electronic lexicon databases by various publishers of thesaurus and dictionaries.

The EDR (Electronic Dictionary Research) project in Japan has produced an electronic dictionary with over 400,000 concepts with mappings to both English and Japanese words (Yokoi, 1995). However, despite the fact that EDR ontology is bigger than that of Cyc, EDR does not provide much detail about constituent concepts. Importantly, EDR demonstrates the power of bilingual directory of lexicons for revealing the semantics of a given sentence and in contrast to WordNet, EDR uses lexicons of two languages. Together with WordNet, EDR assumes that lexical knowledge is the key factor which enables computers to process human languages; both projects however appear to give less emphasis on an important fact, namely, that a meaningful translation between ontologies requires not only lexicon but semantic consideration. The EDR project is a breakthrough, because it operates in the bilingual context of handling of ontology. It should be noted that world knowledge is written in variety of natural languages other than English. In this sense, EDR-like initiative is required for giving access to huge knowledge repository on the Web to individuals regardless of their mother tongue. It is worth mentioning here that Cyc, EDR and WordNet projects use the term ontology to denote a special type of dictionary. In general, of course, ontology is more than a dictionary.

Other problem-domain specific ontological systems include SNOMED (Spackman 2000), where ontology is a large, standardized structured vocabulary of medicine. As another example, the system UNSPSC (www.unspsc.org) provides ontology for products and services (Fridman & Musen 2003).

Table 1 summarises Ontologies of Chat-80, Cyc, WordNet and EDR. The purpose of this comparison is to highlight the expressiveness of ontology in terms of major categories identified and the purpose of each ontological system.

Ontology	**Purpose**	**Categories**	**Implementation**
Chat-80	Natural Language Processing for the Geographical domain	Small ontology with just 20 categories comprising three categories of areas, points and lines at the top.	Prolog-based
Cyc	Huge Expert system spanning over many domains and exhibiting capabilities of Natural Language Processing	Top most categories of things, intangible and individuals, yet there are so many other categories forming many clusters. Largest ontology with more than two millions facts & rules	Lisp-based
WordNet	Comprehensive Natural Language Processing system for English language	Categories include noun, verbs, adverbs, adjectives, and 6 other categories for giving semantics (synonyms, *antonym*, *hyponym*, etc.	Prolog-based
EDR	Bilingual Natural Language processing system between English and Japanese	Includes a dictionary containing mappings between more than 400,000 English and Japanese terms, yet descriptions are less expressive than in Cyc	Prolog-based

Table 1 – Comparison of Chat-80, Cyc, WordNet and EDR

Note that the early versions of these systems have not used object-orientation for representation of ontologies. They were fundamentally written using Prolog or Lisp. Due to the domain specific nature of these systems, they do not provide a general insight into how the computational ontological modelling can be done. However, we appreciate that WordNet and EDR have pioneered new directions in ontology-based natural language processing.

3.2 Ontology in Semantic Web Research

With the emergence of the Semantic Web Project the issue of ontological modelling has become even more important. Future of the World Wide Web is almost certainly the Semantic Web, as defined below.

"Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (Tim Berners-Lee *et al.,* 2001).

The World Wide Web consists of various information sources stored in different formats as texts, structured data, sounds and images on a variety of hardware and software platforms. It has been commonly agreed that Semantic Web must be powered by ontology and there are numerous projects going on the development of relevant technologies. It is also generally recognized that expressiveness of markup languages should be improved for computers to handle semantics or the meaning associated with web resources. The Semantic Web researches aim at making sense of the web resources through ontologies. Most of the approaches to the development of ontologies for the Semantic Web have been based on various extensions to XML like Web page markup languages. XML is the best available language for extending syntax and semantics of web pages developed using standard HTML. Firstly, XML can be used to annotate web pages to incorporate various media types and link up with all types of data sources. Secondly, various relevant technologies, including parsers, style sheets, protocols, object-oriented integrations techniques and database connectivity techniques have been developed to work with XML technology. Thirdly, XML is an Open Source and platform independent. However, it should also be noted that, by default, XML is not capable of representing semantics, but can be easily extended to capture semantics. Examples of XML-based languages include Ontology Exchange Language (XOL), Resource Description Framework (RDF) and Resource Description Framework Schema Language (RDFS). In general, these languages use XML syntax and define the semantic with the help of extra tag names.

XML, RDF, RDFS and ontology languages form a layered architecture for enabling the services on the Semantic Web (Pan & Horrocks, 2003). The bottom level of the architecture is implemented by XML technology, which describes the syntax of resources. The RDF works on top of XML as simple metadata language, which describes Web resources and relations between them. RDF provides a simple semantics for the data model. One layer above the RDF is RDFS, which provides vocabulary for describing properties and classes of RDF resources. The ontology layer of the Semantic Web is defined on top of the RDFS layer. More powerful and comprehensive ontology languages such as OIL (Horrocks *et al*., 2000), DAML+OIL (McGuinness *et al*., 2002) and OWL (McGuinness & van Harmelen, 2004) stand on top RDFS.

Let us discuss some of the key features of extensions to XML technology. XOL (Karp *et al.*, 1999) served as the basis for powerful ontological modelling languages such as OIL, which we will discuss below. RDF (Lassila & Swick, 1999; Frank & Eric, 2004) is also fundamental to Semantic Web and provides a means for expressing semantics of documents without making any assumptions on the structure of the document. RDFS (Brickley & Guha, 2000; Brickley & Guha, 2004) provides basic type schema such as objects, classes and properties for RDF thereby making RDF more descriptive and comprehensive. RDFS is also considered as very limited language for defining ontologies on the Semantic Web. Simple HTML Ontology Extensions (SHOE) is also a well known markup language to define ontologies

(Heflin & Hendler, 2000) by annotating web pages. SHOE comes within the family of extension to XML for ontological modelling for Semantic Web; it allows users to specify extensible vocabularies, defined as ontologies, which consists of category and relations that can be augmented with additional axioms. In addition, Open Knowledge Base Connectivity (OKBC) is an application program interface enabling distributed ontology repository architecture (Chaudhri *et al.*, 1998), which is not XML based; it defines ontology on the basis of frames, which are described by slots, classes (set of entities), individuals (an entity) and facets (properties associated with slots).

The US Defence Advanced Research Projects Agency (DARPA) has also contributed to Semantic Web development by introducing a DARPA Agent Markup Language, known as DAML (Hendler & McGuinness, 2000). The DAML makes the web contents more accessible and understandable (McGuinness *et al.*, 2002). DAML and OIL (Horrocks *et al.*, 2000) have provided the basis for the Web Ontology Language (OWL), which is the starting point for the W3C standards for Semantic Web. The OWL (McGuinness & van Harmelen, 2004) extends the RDF and RDF Schema by introducing additional vocabulary and formal semantics for describing properties and classes, including relations between classes such as disjointedness, cardinality and equality.

It is important to mention that OWL is not only for presenting semantics of contents to humans but also for supporting the processing of contents by various applications. As such OWL aims to enable machine-based semantic manipulation on the Semantic Web thus the OWL initiative is an important contribution to realizing the aim of Semantic Web. In this sense, OWL-S (Ontology Web Language for Services) has been developed as a major approach to extend OWL for supporting the ontological needs of Web services. OWL-S provides three interrelated sub-ontologies, namely, profile, process model and groundings to support web service applications (Martin et al 2004). At this point it is worth mentioning that there are several research projects aimed at translating between Web Service description languages and OWL-S, formerly DAML-S, e.g. http://www.daml.ri.cmu.edu/wsdl2owls, Paolucci et al, 2003. We discuss ontologies for Web Service in Section 3.4.

A variety of tools, platforms and development environments for ontology creation, management and sharing on the Semantic Web are based on the above mentioned base technologies such as OWL, RDF, RDFS and DAML. At present, a good review paper on comparison of above technologies from an ontological viewpoint is lacking. Since many tools have the same XML based grounding, we limit our discussion to major products such as Ontolingua, OIL, and DAML+OIL.

Gruber (1993) has reported on the development of Ontolingua with a set of portable ontologies as a means for sharing and reuse of formally represented knowledge. Ontolingua deals with ontologies represented in predicate logic, frames and relational languages. The ontology of Ontolingua defines forms with classes, relations, functions, objects and theories. Ontolingua ontologies can be translated into different languages including KIF (Genesereth & Fikes, 1992), Loom (MacGregor, 1991), Epikit (Genesereth, 1990) and EXPRESS (Spiby, 1991). Using Ontolingua the same ontology can be used for different purposes with the help of KIF, Loom, Epikit and EXPRESS. For example, Loom can be used for conceptual design of the ontology and managing the knowledge base of facts and objects. Epikit provides explanatory reasoning. EXPRESS is the standard language for logical database design for sharable data. KIF provides ontologies for sharing data. KIF representation is based on predicate logic, but uses a Lisp-Oriented syntax. How Ontolingua works with KIF, Loom, Epikit and EXPRESS is beyond the scope of this paper and an interested reader may refer to Gruber (1993) and (Farquhar *et al*., 1997) for further details. However, we would like to give more details about Ontolingua with reference to its architecture.

Perhaps the best way to elaborate more on Ontolingua is to highlight the features of the recent development of web integration of Ontolingua. This development provides a Web interface for Ontolingua and an Ontolingua server (Farquhar *et al*., 1997) enabling users to publish, brows, create and edit ontologies stored on the Ontolingua server. With these new facilities the Ontolingua is becoming the best-known environment for building specific types of ontologies. The new development of Ontolingua comprises four modules referring to, Ontology Editor/Server, Remote Collaborators, Remote Applications and Stand-alone Applications. The ontology editor/sever provides facilities creating new ontologies and reusing existing ontologies by doing various modifications. This is provided through a general purpose web interface. The Remote Collaboration module provides facilities for remotely distributed groups to build, maintain, share and store Ontologies at the Ontolingua server. The remote application module of Ontolingua provides facilities to use ontologies by querying and modifying them. Modified ontologies can also be stored on the server. The Stand-alone Applications module allows the user to translate an ontology into a specific application. As we mentioned earlier, the fundamental idea behind the Ontolingua is to enable translating and sharing of ontologies. Therefore, stand-alone module is the core of Ontolingua.

Although Ontolingua can translate among limited number of ontological systems (KIF, EPIKIT, LOOM and EXPRESS), it is a relatively bulky product (Horrocks *et al.* 2000). This makes the product more expensive in computational terms. Further, until a framework for the development of ontologies for a wider range is supported and translation of Ontolingua ontologies into a considerable range of ontological systems is enabled, its usage will be very limited. In its present state Ontolingua cannot be used as a general system to support ontology translation and sharing for the Semantic Web application. Nevertheless, the Ontolingua approach can be considered as a good initiative and the same philosophy can be extended towards development of a more general approach to translating ontologies across various applications. It is always more practical to move from specific systems to generic ones than vice versa.

OIL (Ontology Inference Layer) supports ontologies based on the new web standards such as XML, RDF and RDF Schemas (Horrocks *et al*., 2000). OIL is powered by ideas from three disciplines, namely, Description Logic, Frame-based Systems and Web Standards. Description Logic is a powerful class of logic-based knowledge representation, with high expressive power. It describes knowledge in terms of concepts and role restrictions that are used to derive classification from taxonomies. Its description logic also closely matches with frame-based representation (Horrocks *et al*., 2000).

OIL provides three-layered ontology, the layers being Object Level, First Meta Level and Second Meta Level. The object level describes concrete instances of ontology, and this level is strongly associated with XML schema. The First Meta Level defines the actual ontology with well-formed semantics. The Second Meta Level is concerned with describing features of ontologies, in terms of meta-data elements such as Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage and Rights. The choice of this ontology is based on Dublin Core Metadata Element Set Standard (Dublin Core). The actual role of OIL is concerned with the first and the second Meta levels. OIL is strongly bounded by RDF, which is the W3C standard for representing meta-data on the Semantic Web. OIL has also improved RDF to be able to define vocabularies for authoring data by way of RDF-schema specification. Very restricted form of classes in XOL and OKBC are also extended by OIL. In particular, OIL has extended the expressiveness of slots in XOL and OKBC. Therefore, OIL is becoming a base technology rather than a specific product. In contrast to Ontolingua OIL has a very simple and limited core language.

As mentioned earlier, DAML is a Semantic Web markup language developed by DARPA, which works well with OIL. The combined system (DAML+OIL) has been accepted as a starting point for Semantic Web W3C standard. The initial version of DAML was known as DAML-ONT and was released in October 2000. The first release was basically an extension to RDF Schema vocabulary without a powerful mechanism for reasoning. At a later stage, Richard Fikes and Deborah McGuinness developed First Order Logic (FOL) semantics for DAML-ONT, which gave it a more precise foundation for semantic analysis. As powered by FOL, the new ontology language DAML+OIL enables the use of traditional theorem provers and problem solvers, which are based on FOL, to answer queries. All traditional inference engines in Artificial Intelligence are based on FOL.

Modelling Technology	**Modelling Primitives**	**Remarks**
XML	Tags for structuring documents	Cannot define semantics Provides basic technology
RDF	Objects and relations	Uses XML syntax, also provides simple semantics
RDFS	Vocabulary to describe classes and properties of RDF	Adds more semantics to web resources
OWL	Enhanced vocabulary to RDFS with additional relations such as disjointedness, enumeration cardinality, equality, symmetry,	Handles more semantics Intended for processing by machines Intended for interoperability among applications
Ontolingua	Based on predicate logic, frames and relational languages giving emphasis on classes, relations, functions, objects, theories	Supports ontology translation among limited ontological systems (e.g. KIF, LOOM, EPIKIT, EXPRESS) Web accessibility is enabled through Ontolingua server
OIL	Based on frames, description logic and web standard (RDF, RDFS)	Strongly bounded by XML and RDF Core system is simpler and more powerful than Ontolingua
DAML	Similar to OWL	Earlier version of OWL
DAML+OIL	Based on primitives of OIL and DAML	Major starting point for Semantic Web Standards

Table 2: Comparison of some Semantic Web ontological modelling environments

Table 2 summarises the modelling primitives of XML, RDF, RDFS, OWL, Ontolingua, OIL and DAML, with a remark on their underlying basis of definition of ontology. It appears that most approaches are based on object-oriented way of representing ontology or traditional logic-based approaches and frames in Artificial Intelligence. Further, since XML, RDF, RDFS and ontology languages such as OWL forms a layered architecture, they are based on the same fundamental concepts. When ontologies are developed on the basis of same concepts such as frames, logic or object-orientation, it is quite natural to expect their interoperability on the Semantic Web. However, this has not been achieved in reality. As we point out later, the reason for this issue is closely associated with the lack of philosophical understanding of ontologies.

3.3 Ontology in Multi-Agent Systems Research

Multi-Agent Systems (MAS) is yet another fast growing area in Computer Science and promotes distributed problem solving through interaction among agents running on heterogeneous hardware and software platforms. MAS is considered as a new approach to intelligent computing, which is based on the fundamental principle of *emergent intelligence* (Rzevski 2004) or *intelligence through interaction* (Wooldridge, 2002). Ontological modelling plays an increasingly important role in implementing effective communication and sharing of problem solving knowledge in MAS, because ontologies provide a structure which can be comprehended by all interacting agents. It is commonly agreed that ontology works as an artefact for facilitating the interaction among different persons or agents. Ontology in MAS necessarily contains conceptual knowledge which agents consult when making decisions. Typically the whole agent swarm shares ontological knowledge of the problem domain. Domain ontology in MAS is expected to evolve in presence of new experience of agents in the swarm. Agents in a MAS use ontologies for communicating with the agents within and outside the swarm.

Current research into MAS is predominantly concerned with Agent Communication Languages (ACL), such as KQML, and Capability Description Languages (CDL), such as DAML-S (Ankolekar *et al*., 2002) and LARKS (Sycara *et al*., 1999) for defining ontologies. Knowledge Query Manipulation language – KQML (Finin *et al*., 1994) is the standard approved by the Foundation for Intelligent Physical Agents (FIFA) for Agents communication. As we discussed earlier, DAML-S is the Ontology description language introduced as a part of DARPA funded project on Semantic Web. It should be noted that MAS development is a huge process and we limit our discussion only to the ontological modelling aspects of MAS.

Ontological modelling support is an essential feature of MAS development environments in general. All such environments support the object oriented ways of modelling ontologies in terms of classes, objects and properties. For example, JADE (http://jade.tilab.com) provides reusable ontologies as a class *jade.content.onto.ontology*, which can be extended to develop new ontologies (Caire, 2002). Ontology development in JADE is closely linked to Java programming. In contrast, MAS development environments such as Magenta Multi-Agent Platform (http://www.magenta-technology.com/products/platform.shtml) and Zeus (http://www.labs.bt.com/projects/agents/zeus) provide attractive graphical tools for ontological modelling. In particular, as Wooldridge (2004) pointed out Magenta Ontology Management toolkit provides support not only for creation of Ontology but also for push-button translation of ontologies to J2EE (Java 2 Enterprise Edition). This is an important feature of Magenta Technology, because many MAS development tools provide facility for creation of ontology, leaving MAS developers to program the linking of ontology with other components of MAS. Further, ontology created by Magenta can be exported in many other forms, such as XML. Magenta Ontology can be shared and reused by developers across projects, while this is not possible in Zeus at the moment. Further, based on their survey on MAS platforms, Ricordel & Demazeau (2000) point out that although Zeus provide graphical support for reuse of ontologies, adding new functionality requires return to Java code. Therefore, Zeus cannot be identified as a tool for novice MAS developers.

Magenta Multi-Agent Platform and Ontology (Rzevski & Skobelev, 2004) and (Batishchev, Iwkushkin, Minakov, Skobelev and Rzevski, 2001) has a very wide application scope and powerful visual editors and has been successfully applied to real world applications such as car production planning and scheduling, management of a fleet of trucks and very large tankers, and planning of complex logistic networks for the transportation of heavy loads involving airships, barges, cargo aircrafts and sea going vessels.

While ontological modelling is an integral part of MAS development, some MAS toolkits, for example, Agent Academy (Mitkas *et al.*, 2003) do not provide their own tools for ontological modelling, instead they use standard environments such as Protégé 2000. The Agent Academy provides facilities for drastically reducing programming activity as compared with JADE and similar MAS development environments. The use of standard tools for ontology development in MAS is becoming more practical since there are many comprehensive general tools available for ontological modelling. We report on such tools in section 3.6.

Other toolkits such as Jack (http://www.agent-software.com.au) and MadKit (http://www.madkit.org) can be considered as MAS programming environments. These tools have not placed a particular emphasis on ontological modelling. Persons with Java programming knowledge can use these tools to learn about MAS development. In general MAS development environments have been designed to run on Java platform. This is a very rational choice, since MAS should be able to run in a platform independent manner in heterogeneous environments. Also, Java is an Open Sources resource and one can do various experiments and development in Java free of charge.

There are also a number of methodologies for analysis and design of MAS, e.g. GAIA (Wooldridge *et al*. 2000), PASSI (Cossentino & Potts, 2002) and Agent.Enterprise (Stockheim *et al*. 2004). These methodologies basically provide object-oriented guidelines for design of MAS. PASSI and Agent.Enterprise place also a particular emphasis on ontological modelling as a part of MAS development. Recently, UML extensions have been proposed as Agent Modelling Languages (standard UML notation is adequate only for modelling of standard object-oriented systems, but not for MAS-like specific systems). In this sense, AgentUML (Odell *et al.*, 2001) is the best known language that extends UML notation to accommodate modelling constructs of Agent technology. MAS development CASE tools such as PASSI (Cossentino & Potts, 2002) also use UML extensions in their methodologies. PASSI strongly support ontological modelling. In fact, PASSI provides tools for the development of domain ontology as well as communication ontology. This is a significant feature of PASSI, because most of MAS tools provide support only for development of domain ontology ignoring ontology of communication. Research into developing ontology for agent communication is reported by Tamma *et al*. (2002) and Aart *et al*. (2002).

One of the most important developments in Multi-Agent Systems research is the investigation into mental concepts such as BDI - Belief, Desire and Intention (Rao & Georgeff, 1991;

Wooldridge 2000; Ferrario & Oltramari, 2004). However, there seems to be no major research into the use of mental concepts for ontological modelling, though mental aspects have been identified as important categories of ontology. For example, most cited Cyc ontology also identifies beliefs, perception, etc. as categories within the cluster of sentient beings (http://www.cyc.com/cyc/technology/whatiscyc_dir). Further, although some tools use BDI models for design of agents, there are still ambiguities in the use of terms such as "rational" and "intentional". For example, Ferrario & Oltramari (2004) state that autonomy, reactivity, pro-activity and social ability are features of intentional agents. In contrast, according to Wooldridge (2000), agents with such features are referred to as rational agents. Further, Russell & Norving (2003) define rational agents as the agents which do the right thing. Such ambiguities exemplify the need for research into the use of mental factors for ontological modelling. This is why, in section 4 of this paper, we emphasise the need of in-depth study of mental phenomena for the establishing new directions in ontological modelling. We are of the opinion that mental factors can be expressed in a rudimental fashion in RDF like XML related technologies. In fact in Section 5, we show how XML technology could be used to incorporate mental factors into ontological modelling.

Tools/Methodologies	**Ontology modelling support**	**Remarks**
JADE	Provides ontological modelling support	Using Java programming ontologies; can be reused and extended
Magenta	Graphical tool support for ontological modelling and integration	Automated process of linking ontology with other component of MAS; ontologies are reusable within but not across projects
Zeus	Graphical tool support for ontological modelling and integration	Linking ontology with other components in MAS is a manual process; ontologies are reusable within but not across projects
MadKit	No particular support for ontological modelling	A very good simple tool to learn about development of MAS
Jack	No particular support for ontological modelling	More like a programming language for the development of MAS
GAIA	No particular support for ontological modelling	Mainly supports analysis and design of MAS; ontological modelling is lacking
PASSI	Strong ontological support for various purposes	Both domain knowledge and communication specifications are viewed as ontology

Table 3: Comparison of tools/methodologies for ontological modelling in MAS

Ontological modelling is considered in MAS literature only as an integral part of MAS development. There is no comparative research of ontological modelling capabilities of various environments for MAS development. However, there are some areas of MAS, where ontological modelling has been a main theme. These areas include Middle Agent (Genesereth & Ketchpel, 1994; Payne *et al.* 2002b; Lu (2004), Ontology Translations and the recent papers on Web Services. From ontological viewpoint, the whole idea behind the Middle Agent is based on the importance of sharable ontologies among various applications.

Since MAS is a vast area of research, a comparison of tools and methodologies for developing MAS can be done from various viewpoints. Table 3 illustrates comparison of some major tools and methodologies from the viewpoint of their support for ontological modelling in MAS.

Note that some tools are basically programming environments (e.g. JADE, MadKit and Jack) while others (e.g. Zeus, Magenta, PASSI) provide computer-aided support for MAS development with an additional component of ontological modelling. There are also tools/environments, which do not give a particular emphasis on ontological modelling (e.g. GAIA, MadKit and Jack). It should be noted that the purpose of the section on Ontology in MAS is not to provide an exhaustive list of tools and methodologies, yet to show their diversity in terms of ontological modelling.

3.4 Ontology in Web Services Research

The Web Services is considered as a fast growing area of research. Fundamentally, Web Services should be able to locate other services that can provide solution to problem at hand and compose a solution by interoperating complex services (Paolucci et al., 2002). Among others requirements, Web services are expected to enable the tasks of automatic web service discovery, automatic web service invocation, automatic web service composition and interoperation and automatic web service execution monitoring (Ankolekar *et al.*, 2002). Until recently, Web Services was considered as a separate area of research (Paolucci *et al.* 2002). Web Services had their own description languages such as UDDI (Universal Description Discovery and Integration) and WSDL (Web Services Description Language). UDDI provides a registry of business and services in terms of attributes such as name, address and service that they provide, and classification of services within taxonomies. However, UDDI does not describe service capabilities and this is the role of WSDL. The WSDL descriptions also provide a guidance how to use services (e.g. operation sequencing, sate management), obligations of participants, compositionality of results, etc. (Paolucci *et al.* 2003).

Many researchers (e.g. Paolucci *et al.,* 2002; Korhonen *et al.*, 2002) have identified MAS as a technology which can be used to improve Web Services as well as Semantic Web, and therefore the demarcation between these disciplines is gradually disappearing.

The convergence of the three disciplines is best illustrated by the following considerations. As Paolucci *et al.* (2002) point out semantic matching requirements between service advertisements and service requirements in Web Services applications cannot be adequately supported by languages specifically developed for that purpose, namely UDDI or WSDL yet DAML-S, developed for Semantic Web applications is perfectly adequate; DAML-S is, in fact, a general language for describing semantics of ontologies and therefore applicable in all areas where ontology is the key system component. A very good comparison of capabilities of UDDI and WSDL with DAML-S can be founded in Ankolekar *et al.* (2002). We have reported research on translation of WSDL to DAML-S in subsection 3.6.7, which is placed under the section on General environments for ontological modelling.

LARKS (Language for Advertisement and Request for Knowledge Sharing) is yet another language for capability description of agents (Sycara *et al.,* 1999) which has a general applicability. Firstly LARKS is an approach for implementing web services dealing with requests and advertisements. Secondly, it can also be considered as a matchmaking approach to communication between incompatible agents. The matching engine of LARKS includes several filters for context matching, profile comparison, similarity matching, signature matching and constraints matching. These filters progressively restrict the number of advertisements to be matched. Sycara *et al*., (1999) claim that LARKS is capable of both syntactic and semantic matching with the user intervention. LARKS uses frame-based structure of ontology with a relatively small set of slots. Continuing our discussion on convergence, we note that Korhenon *et al.* (2002) have used DAML+OIL for developing workflow ontology in MAS in the context of Web Services. Among others, UML has also been extended to design of ontologies for MAS (Bergenti & Poggi, 2001). It appears that there is a certain overlap among different Web Service standards and languages while groundings are quite similar in terms of object-oriented concepts, frames, XML technologies, etc. It is obvious that standards must be developed to support further progress in this area.

3.5 Translation of Ontologies

Different ontologies may not share the same vocabulary even for referring to the same entities. For example, an agent searching for a book may use the term *book* in its ontology, while an agent dealing with the inventory of a bookstore may use the term *item* to refer to a book. This simple example shows that there is a need for translation between ontologies belonging to different systems or problem domains before making any attempt at machine interpretation of meaning. Ontology translation is essential even when two software systems seek to support interoperability using DAML or OWL-like standards (Burstein, 2002). This is because such languages are designed for ontology description but not for translations of ontologies across domains.

Dou *et al*. (2002) have introduced, what they called, *Ontology Merging* approach as a means of translating ontologies. The merging of two related ontologies is achieved by taking the union of terms and axioms. Then the terms in two ontologies are related by *bridging axioms*. Dou and colleagues claim that merged ontology works not only as a bridge between two ontologies but also as a new ontology for further merging with other ontologies. This point has a great importance since ontology should allow us to evolve domain knowledge, or at least, think beyond the current ontology content. Dou *et al*. (2002) have deliberately separated syntactic translation from semantic translation of ontologies. Syntactic translation is automated through the translator PDDAML, while semantics translation expects intervention from human experts. Separation of syntactic translation from semantics translation is philosophically important as these two aspects deal with different levels of complexity. In general syntactic translation is more straightforward than semantic translation.

It should also be noted that the term “merging ontologies” has been treated differently by different researchers. For example, PROMPT (Fridman & Musen, 2000, 2002, 2003) and Chimaera (McGuinness *et al*., 2000) treat ontology merging as a kind of editing of two related ontologies. PROMPT has been developed on the basis of Protégé 2000 (Fridman *et al*., 2000), which is a well-known knowledge modelling environment for ontology design and knowledge acquisition for knowledge based systems. Protégé 2000 is an ontology editor and PROMPT works as a Plug-in for Protégé 2000. Chimaera works the same as PROMPT, but Chimaera has a limited set of ontology merging tools. Based on their evaluation, Fridman & Musen (2000) report that PROMPT had 30% more correct suggestions than Chimaera did, when applied to the same source of ontologies. Further, PROMPT gives more specific instruction than Chimaera to the user for merging ontologies. According to Dou *et al*. (2002), the ontology merging in PROMPT and Chimaera require user intervention. Further, PROMPT and Chimaera have not treated syntactic and semantic translations as two aspects to be worth considering separately.

It is quite clear that semantic translation of ontologies is a crucial task, which currently requires human intervention. Until we implement semantics processing of ontologies by machines, we cannot claim that the goals of the Semantic Web Project have been achieved.

3.6 General Environments for Ontological Modelling

Thus far we have discussed ontological modelling environments from different viewpoints including specific ontologies and ontology research in Semantic Web, MAS and Web Services. Now we discuss ontology modelling environments from a more generic perspective. The environments discussed below can be used as general approaches to the development and manipulation of ontologies for various purposes. Perhaps the OntoWeb (2002) survey of ontology tools can be considered as the most comprehensive survey with regard to ontology development environments. There are also Web resources such as multi-agent.com and Carnegie-Mellon University website, which provide information about various tools and environments for ontological modelling. These resources have also been classified into specific categories including Semantic Web, Multi-Agent systems and Web Services. The purpose of this section is to discuss ontological modelling tools/environments in a more general manner without referring to a particular area of application.

OntoWeb survey has classified ontology modelling environment into six generic groups, namely, o*ntology development tools, ontology merging and integration tools, ontology evaluation tools, ontology-based annotation tools, ontology storage and querying tools*, and o*ntology learning tools*. Each group has been evaluated against different criteria. We do not intend to discuss in detail the OntoWeb study, yet we make use of its results to point out the various tools available and some of their key features. We also wish to add a subsection on *ontology translation* tools at the end of this discussion. We point out that this section is another dimension of ontology tools, which is not covered by OntoWeb like surveys.

3.6.1 Ontology Development Tools

OntoWeb has identified 11 environments, namely, Apollo, LinkFactory, OILEd, OntoEdit, Ontolingua, OntoSaurus, OpenKnoME, Protégé 2000, SymOntoX, WebODE and WebOnto for building or development of ontologies. All these are graphical tools, based on object-oriented concepts and various extensions to XML including RDF, RDFS, OIL and DAML+OIL. Most tools are developed using Java, while some have chosen C++. Further almost all tools provide a library of ontologies. Most tools have some means of working with other Ontology environments, which support interoperability among tools. As OntoWeb (2002) survey reveals none of the tools provide mechanisms for exception handling. Further, LinkFactory, OILEd, OntoSaurus and OpenKnoME are the only ones with reasonably good ability for automatic manipulation of classification of ontologies. According to the survey, Protégé 2000 has been the most widely used ontology development tool and it has provided a basis for the development of other powerful tools such as PROMPT; Protégé 2000 enables linking with dictionaries such as WordNet. Further, Protégé 2000 promotes reusability of ontologies in association with Ontolingua server library (Pinto *et al.*, 2002). A comprehensive guide for using Protégé 2000 is given by Fridman & McGuinness (2004).

3.6.2 Ontology Merging and Integration Tools

Ontology merging and integration tools generally come as components within ontology development environments. For examples, well known ontology merging tools, namely, PROMPT and ODEMerge are integrated within Protégé 2000 and WebODE, respectively (OntoWeb, 2000). In general, ontology merging tools are expected to be interoperable in terms of different languages and development environments. OntoWeb (2000) has identified Chimaera, FCA-Merge, PROMPT and ODEMerge as major tools for merging ontologies. All these tools are interactive, while ODEMerge is said to be fully automated. However, as Dou *et al*. (2002) pointed out the automatic translation among ontologies is too ambitious, especially in the context of semantic translation. In general, ontology merging tools are necessarily ontology editing tools. Ontology translation tool must be able to support not only all the development environments that are based on technologies including frames, first-order logic, XML extensions and object-orientation but also various data models on the Web.

3.6.3 Ontology Evaluation Tools

Ontology evaluation is a relatively new and challenging area because evaluation criteria for ontologies are not commonly agreed. According to OntoWeb (2002), Ontology evaluation tools are developed for assessing *Ontology properties* and *technology properties*. OntoWeb (2000) discusses OntoAnalyser, ONE-T and OntoClean as environments which can evaluate Ontology properties, in particular language conformity and consistency. OntoGenerator is singled out as a tool for the evaluation of technological properties, in particular, performance and scalability. These tools are useful for maintenance and management of ontological systems. However, since the field of ontology development is not yet adequately matured, ontology evaluation may be a premature concept. Ontology evaluation tools must be developed to be able to comprehend ontological tools, which are contracted on the basis of various technologies.

3.6.4 Ontology Annotation Tools

Ontology annotation tools allow the use of a pre-existing ontology to insert markup into web pages or other documents such as a local file. All such tools are necessarily based on XML and its extensions such as DAML and OIL. OntoWeb has identified five annotation tools, namely, AeroDAML, COHSE, MnM, OntoAnnotate, OntoMat and SHOE. These tools have different emphases, for example, MnM aims at populating ontologies, while OntoMat, OntoAnnotate and COSHE aim at knowledge markup within web pages. AeroDAML is specific to web pages containing DAML markup. Annotation tools can generally be used by persons with little programming knowledge.

3.6.5 Ontology Storage and Querying Tools

Ontology storage and querying tools perform the roles of storing, managing and enabling querying on ontologies. These tools provide somewhat similar features as database management systems and respective query processing environments. However, ontology modelling primitives in RDF like standards are considerably different from those defined in traditional database models such as objects, relations, facts, etc. This reason has lead many researches into ontology storage and querying tools. OntoWeb (2002) has classified those tools into three categories on the basis of three standards, namely, RDF, DAML+OIL and Topic Maps. Discussing features of these tools is beyond the scope of this paper because they do not concern ontological modelling.

3.6.6 Ontology Learning Tools

Ontology learning tools are designed to automatically derive ontologies from natural language text documents. Hasti (Shamsfard & Barforoush, 2004) describe an automatic ontology construction system, which operates on Persian text documents. These systems are much more complex than ordinary ontology modelling tools due to their additional task of natural language processing. Examples include: Cyc (Lenat, 1995), TEXT-TO-ONTO (Maedche & Staab, 2001) and SynDiKATe (Hahn & Romacker, 2000). The ontology learning systems differ from each other based on factors such as learned elements, starting point, pre-processing, learning methods, resulting ontology and evaluation method (Shamsfard & Barforoush, 2004). According to Shamsfard & Barforoush, Hasti is close to TEXT-TO-ONTO as both are capable of learning concepts, taxonomies, and non-taxonomic relations. Undoubtedly, realising the aim of the Semantic Web project requires technologies for ontology construction from various sources on the Internet.

It appears that almost all ontology development environments have the same grounding. Therefore, developing yet more environments leads to duplication of products with minor variations. One way to overcome this drawback is to integrate similar systems into comprehensive environments. This approach has been taken by many researchers. Among others, ZEUS (Nwana *et al.*, 1999) MAS toolkit is a creative synthesis of already invented elements. ZEUS provides a visual ontology editor together with set of built-in components for the development of MAS. The ZEUS toolkit has been successfully used for developing MAS for various applications including commercial workflow management systems.

3.6.7 Ontology Translation Tools

Until recently independent research fields, Semantic Web, Multi-Agent Systems and Web Services, each with their own ways of representing knowledge, began to converge. As a result, various research projects were initiated with the aim of developing ontology translators. Examples include translators between the fields of Web Services and the Semantic Web, including ones from WSDL (Web Service Description Language) to OWL-S (formally DAML-S). Using WSDL2DAML-S grounding ontology of DAML-S can be completely generated. However, other two ontologies, i.e., process model and profile model can only be generated partially by the WSDL2DAML-S due to differences in information in WSDL and DAML-S (Paolucci et al, 2003). Despite basic limitations at this stage, it is said that WSDL2DAML-S translation saves a great deal of manpower. For example, Paolucci and others say that translation of complex WSDL documents specifying Amazon Web services takes about one week, while WSDL2DAML-S takes few hours. Complete and automated ontology translation among different applications, that use various data formats, must be achieved to realise the expectations of Semantic Web.

Table 4 summarises various categories of ontological tools with an emphasis on their technological basis for ontological modelling and provides remarks on each category. Note that the aim of comparison is to highlight the differences in approaches to ontological modelling rather than assess the usability of various tools.

It is evident from Table 4 that success of ontological tools is heavily dependent on their ability to cope with multiple development techniques and interoperability among various applications. Most tools are developed on the basis of same technologies. The roles of some types of tools such as evaluation tools, merging and integration tools are yet to be achieved. Ontology learning and translations are relatively new areas of ontological modelling.

Tool Type	Modelling basis	Remarks/usage
Development tool	XML, RDF, RDFS, OIL, DAML, OWL-S Frames, First order Logic, Object orientation	Supports the development of ontology for any purpose; may be introduced as an add-on to other systems
Merging & integration tool	In theory, should be able to support all types of modelling techniques	Supports interoperability and editing of ontologies; presently the semantic handling by machines is a key issue
Evaluation tool	In theory, should be able to support all types of modelling techniques	Currently stress is placed on accessing the ontology properties and technology properties
Annotation tool	HTML, XML, DAML+OIL	Work as a knowledge markup for existing web pages; Used for populating ontologies
Storage & querying tool	RDF, DAML+OIL, Topic Map	Supports ontology storing, managing and querying; does not support ontology development
Ontology learning tool	In theory, should be able to support all types of modelling techniques	Derives ontologies from various natural language texts
Translation tool	In theory, should be able to support all ontological modelling technologies and other data formats	Enables translation of ontologies among various heterogeneous applications

Table 4: Comparison of categories of ontological tools

4 ISSUES

In spite of intensive research effort, there are still important issues in the area of ontological modelling that need to be resolved. Before we point out the key issues that we have identified in our research, first we present some of the major ontological issues identified by other researchers.

Shamsfard & Barforoush (2004) have reported three issues, namely, lacking standards to integrate and reuse existing ontologies; using fixed categories based on single viewpoint; and absence of full automatic knowledge acquisition methods.

Dou *et al*. (2002) have concluded that the key issue is semantic translation; the need for human intervention to establish the meaning of ontological content has affected the entire filed of ontological modelling, in particular ontology merging and translations.

OntoWeb (2000) identified four issues: (a) although there are many “similar” tools, they neither interoperate nor do they cover all activities of the ontology development life cycle; (b) ontology merging and integration has not been explored sufficiently; (c) standards must be developed for ontology evaluation and (d) there is a need for research into developing a common workbench for ontology developers and a set of ontology middleware services.

It is quite clear from the reports by others, e.g., Shamsfard & Barforoush (2004) and OntoWeb (2000), as well as from our own research that most tools/methodologies for the development of ontologies are based on extension to XML (RDF, RDFS, DAML, OWL, etc.), frames, logic or object-oriented concepts. It is therefore rather surprising to note that, though, most ontology development tools/environments use the same viewpoint, the automated machine operability of ontologies is still a problem. Maybe there is something important missing from the ontological viewpoints that we are holding at present. Maybe there is a need for rethinking of ontological modelling from a different philosophical viewpoint.

We identify the following as two key issues which need to be addressed.

- At present, Ontology developers impose on Ontology users their own worldview; this means that communities of users consisting of individuals with different backgrounds, different experiences and different skills have to accept conceptual knowledge about a domain as given to them by Ontology developers
- Ontologies are constructed as permanent edifices of knowledge, to be updated by developers when they deem it necessary; no mechanism for the evolution of knowledge is designed into Ontology management systems

These issues are fundamental. We argue that the ability to design computer readable ontology which adequately reflect worldviews (or perception) of their users and which evolves as knowledge of their users grows, would lift the whole area of ontological modelling to a different level of sophistication.

5 RESOLVING ISSUES

Knowledge accumulated through centuries shows that ontology comprises both mental and physical phenomena. In spite of this, current approaches to ontological modelling in computing have not given a particular emphasis on value of mental factors.

We argue that bringing mental factors into ontological modelling provides a novel way of looking at computational ontological modelling, which can address the two issues identified in the previous section, namely, (a) capturing individual perceptions and (b) supporting ontology evolution.

To bypass any objection to this approach on the grounds that mental factors properly belong to epistemology rather than ontology, let us say that although the demarcation between metaphysics and epistemology is arguable, an appropriate enhancement of ontological modelling naturally contributes to the increase in the epistemological richness of the problem domain. This is because the inclusion of a more comprehensive set of categories into a domain model reveals the dimensions along which epistemological concerns of the domain can also be resolved. Therefore the introduction of mental factors into ontological modelling leads to more comprehensive domain models, regardless of whether we consider mental factors as belonging to metaphysics or epistemology.

We propose a simple way of introducing mental attributes into ontology.

Consider a system in which every individual problem owner is represented in Problem Domain Ontology by the class of objects called Person. This object will have, in addition to various physical attributes, a new class of Mental Attributes, such as Confidence, Motivation, Satisfaction and Preference. Values of these Mental Attributes would be stored together with values of physical attributes in the Attribute Value Database.

Each individual problem owner would be represented by a Personal Agent that has access to both Domain Ontology and Attribute Value Database and, most importantly, a permission to adjust own Mental Attribute Values.

Let us explain how this would work on an example. A Personal Agent, PA1, deals with other agents such as Document Supply Agent (A_1), Information Retrieval Agent (A_2) and Online Bookstore Agent (A_3). The experience of PA1 in dealing with other agents is recorded and used by PA1 to adjust values of its mental attributes such as Satisfaction and Preference; for example, PA1 may have a high level of satisfaction in dealings with A1 but a low level of satisfaction when dealing with A3; his preference may be to send requests for information first to A2 and then to the other two agents. Also, PA1 evaluates its own performance, and uses results of self-evaluation to adjust values of mental attributes such as Confidence; for example, PA1 may have a high level of confidence in its abilities to acquired updates in the area of X and a medium confidence in getting good results when involved in area Y. PA1 will use its level of confidence to estimate the time interval required to get to the user requested information. Further, PA1 assesses the urgency/difficulty of the current problem solving situation and suitably adjusts value of Motivation for the task in hand. The level of motivation influences the selection of task accomplishment strategies; for example, in case of a high urgency of completing a task (high motivation) PA1 may decide to simplify request conditions or drop an information source.

The introduction of the additional category of mental attributes and the establishment of personal agents capable of adjusting own mental values fundamentally changes the system behaviour. Ontology now reflects individual perceptions of system stakeholders.

An additional feature could be easily introduced, namely the support for stakeholders to monitor changes in own mental values and, if deemed necessary, manually adjust them during system operation.

The implementation of mental attributes of Personal Agents and their values into Ontology and Attribute Value Database can be done using XML, as shown in Fig. 1. For simplicity sake we have not included DTD declaration of XML code in this description. The **mental_factor** is the XML root element. For each PA, Ontology of mental factors is defined with respect to child element called, **agent,** which is identified by an agent id.

The XML element, **external_agency** records PA's experience about outside agents, for example A1 and A2, in terms of mental factors such as Satisfaction. The values of mental factors, for instance, x_1 and y_1 could be defined as some qualitative or simple quantitative measures on a known scale (say 1-5). In the usual manner, the XML tag <external_agency> </external_agency> can be repeatedly used to report on any arbitrary number of external agents and the tasks identified by task ids. Figure 1 shows PA's impression of two agents, A1 and A2, when handling the task 101. The task id has been introduced as a key to link up the mental Ontology with the domain Ontology. More specific reference can be made by introducing concepts such as session id within the element of external_agency. Further the XML element, **self-evaluation** represents PA's evaluation of own achievements after handling a particular task. For this purpose, we may consider mental factors such as Confidence. The tag, <self_evaluation> </self_evaluation> can be used to express self-evaluation of a PA regarding each task, which has already been completed. Similarly, the XML element, **current_task** presents the PA's assessment of the task at hand. This may be expressed in terms of mental factors such as Motivation.

```
<mental_factor>
        <agent id ="PA1">
                <external_agency>
                        <task id = " 101 ">
                        <external_agent>  A1 </external_agent>
                        <satisfaction>   x1 </satisfaction>
                        <performance> y1 </performance>
                        </task>
                </external_agency>

                <external_agency>
                        <task id = " 101 ">
                        <external_agent>  A2 </external_agent>
                        <satisfaction>   x2 </satisfaction>
                        <performance> y2 </performance>
                        </task>
                </external_agency>

                <self-evaluation>
                        <task id = " 101 ">
                        <confidence>  a1  </confidence>
                        <competence> b1 </competence>
                        </task>
                <self-evaluation>

                <current_task>
                        <task id = " 102 ">
                        <motivation>  c1  </motivation>
                        <fear> d1 </fear>
                        </task>
                <current_task>
        </agent>
</mental_factor>
```

Figure 1 – XML Elements Describing Mental Attributes of Personal Agents

All PAs maintain their mental ontologies, in association with the domain Ontology, by managing the Attribute Value Database. Each individual PA has access to its own mental Ontology and can change the values of attributes during the PA's life cycle. In principle, mental ontologies of all PAs' can be made visible to all Agents in the swarm. This would allow a cross fertilization of the Agents' perception of the domain. Figure 2 shows the association among PAs, Domain Ontology and the Attribute Value Database.

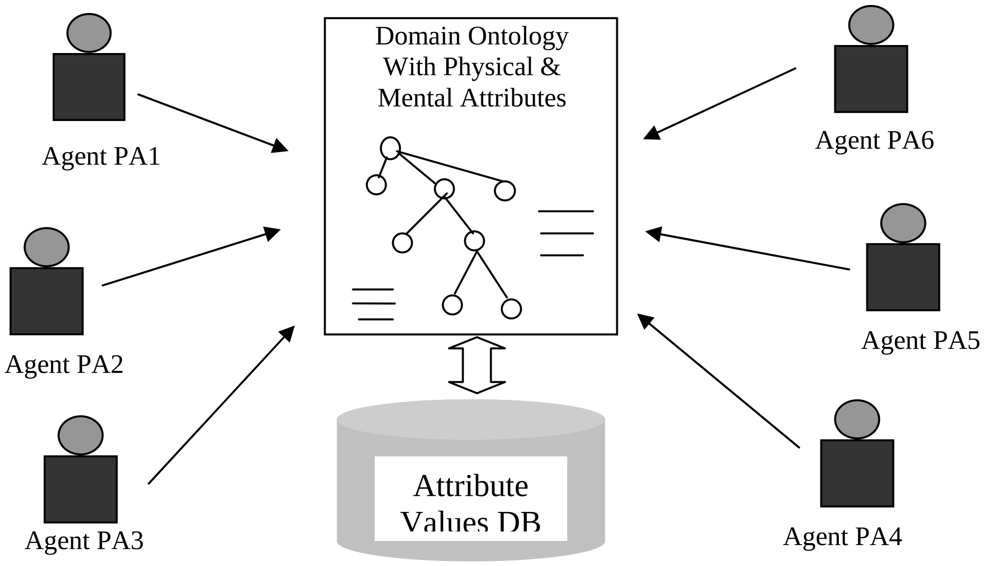

Figure 2 – Domain Ontology with Mental Attributes

From this example it is clear that the addition of Mental Attributes to Ontology is a straightforward process, and yet it may lead to significant results.

6 CONCLUSION AND FURTHER RESEARCH

We have argued in this paper that by revisiting knowledge accumulated over centuries by efforts of many philosophers, we can make a considerable progress towards building "better" problem domain ontologies. By "better" we mean ontologies which are attuned to perceptions of problem owners and capable of evolving as problem domain changes. We also showed how easily mental factors can be incorporated with domain ontologies through XML technology, which is very simple, yet the base technology for all advancements in ontological modelling researches.

Further work will include identification of most appropriate mental attributes and their relationships for the inclusion into ontologies and the development of mental attribute plug-in tools and linking them to standard ontology development environments. At present we are researching into an in-depth study of Eastern and Western philosophical viewpoints on mental factors, and especially the relevance of Buddhist philosophical doctrine which underline the relativity of ontology and the importance of mental phenomena in ontological modelling.

Another challenging and interesting aspect of the further work will be the development of algorithms for agent's own performance evaluation, for assessing satisfaction in dealing with other agents and for judging the urgency of the task at hand.

7 ACKNOWLEDGEMENTS

We acknowledge that this research project has been partially funded by Commonwealth Scholarship Commission in the UK.

8 REFERENCES

Aart, C. van, Pels, R., Caire, G. and Bergenti, F. 2002. Creating and Using Ontologies in Agent Communication, in: *Proceedings of the Second International Workshop on Ontologies in Agent Systems,* 1st International Joint Conference on Autonomous Agent and Multi-Agent Systems, 44-51.

Ankolekar, A., Burstein, M., Hobbs, J. R., Lassaila, O., Martin, D. L., McIlraith, S. A., Narayanan, S., Paolucci, M., Payne, T., Sycara, K. and Zeng, H. 2002. DAML-S: Semantic Markup for Web Services, *International Semantic Web Working Symposium.*

Batishchev, S., Iwkushkin, K., Minakov, I., Skobelev, P. and Rzevski, G. 2001. Magenta Multi-Agent Systems: Engines, Ontologies and Applications, in: *Proceedings of the 3rd International Workshop on Computer Science and Information Technologies CSIT'2001,* Ufa, Russia, 21-26 September, 2001. - Ufa State Aviation Technical University - Institute JurInfoR-MSU, Vol. 1: Regular Papers, 2001, 73-80

Bergenti, F. and Poggi, A. 2001.Exploiting UML in the Design of Multi-Agent Systems, *Lecture Notes in Computer Science.*

Brickley, D. and Guha, R. V. 2000. *Resource Description Framework (RDF) Schema Specification 1.0 W3C Candidate Recommendation*, http://www.w3.org/TR/2000/CR-rdf-schema-2000327.

Brickley, D. and Guha, R. V. 2004. *RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004* http://www.w3.org/TR/2004/RCE-rdf-schema-20040210.

Burstein, M. H. 2002. *Ontology Mapping for Dynamic Service Invocation on the Semantic Web,* American Association for Artificial Intelligence.

Caire, G. (2002). *JADE Tutorial – Applications Defined Content Languages and Ontologies,* TILab S.p.A.

Chaudhri, V. K., Farquhar, A., Fikes, R., Karp P. D. and Rice, J. P. 1998. OKBC: A Programmatic foundation for knowledge base interoperability, *in: Proceedings of the 15th National Conference on Artificial Intelligence, Madison, Wisconsin; and the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI 98).*

Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S. 2001. Web Service Description Language (WSDL) 1.1, *http://www.w3.org/TR/2001/NOTE-wsdl-200110315.*

Cossentino, M. and Potts, C., 2002. A CASE tool supported methodology for design of multi-agent systems, in: *Proceedings of International Conference on Software Engineering Research and Practice,* CSREA Press.

Decker, K., Sycara, K. and Williamson, M. 1996. Matchmaking and Brokering, in: *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS 96).*

Deschaine, L.M., Brice R. S. and Nodine, M. H. 2000. Use of InfoSleuth to Coordinate Information Acquisition, Tracking and Analysis in Complex Applications, in: *Proceedings of Advanced Simulation Technologies.*

Dou, D., McDermott, and Peishen, Qi. 2002). Ontology translation by Ontology merging and Automated reasoning, in: *Proceedings of EKAW workshop on Ontologies for Multi-Agent Systems.*

Dublin Core, http://purl.oclc.org/dc.

Finin, T., Fritzson, R., McKay, D., and McEntire 1994. KQML as an Agent Communication Language, in: *Proceedings of the third International conference on Information and Knowledge Management (CIKM'94), ACM Press.*

Fridman, Noy N. and Musen, M. A. 2000. PROMPT: Algorithm and Tool for Automated Ontology merging and Alignment, in: *Proceedings of the Seventh national Conference on Artificial Intelligence (AAAI-2000).*

Fridman, Noy, N. and Musen, M. A. 2003. The PROMPT suite: interactive tools for Ontology merging and mapping, *International Journal of Human-Computer Studies*, **59**, 983-1024.

Fridman Noy, N. and McGuinness D. L. 2004, *Ontology Development 101: A Guide to Creating your First Ontology,* http://www-protege.stanford.edu/publications/.

Fridman, Noy, N. and Musen, M. A. 2002. Evaluating Ontology-Mapping Tools: Requirements and Experience, in: *Proceedings of the OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002*, 1-14.

Fridman, Noy, N. Fergoson, R. W. and Musen, M. A. 2000. *The Knowledge model of Protégé 2000: combining interoperability and flexibility,* Stanford medical Informatics Technical Report, Stanford University.

Farquhar, A., Fikes, R. and Rice, J. 1997. The Ontolingua Server: a tool for collaborative Ontology construction, *International Journal of Human-Computer Studies,* **46,** 707-727.

Fowler, J., Perry, B., Nodine, M. and Bargmeyer B. 1999. *Agent-based Semantic Interoperability in InfoSleuth,* SIGMOD Record 28, 60-67.

Frank, M. and Eric, M. 2004. REF Primer, *W3C Recommendation,* 10th February 2004, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

Ferrario, R. and Oltramari, A. 2004. Towards a Computational Ontology of Mind, in: *Proceeding of International Conference on FOIS 2004,* IOS Press.

Genesereth, M.R. 1990. *The Epikit Manual,* Epistemic, Inc. Palo Alto, CA.

Genesereth, M.R. and Fikes, R. E. 1992. *Knowledge Interchange Format, Version 3.0 Reference Manual,* Technical Report Logic-92-1, Computer Science Department, Stanford University.

Genesereth, M.R. and Ketchpel, S.P. 1994. Software Agents, *Communication of the ACM*, **37**(7), 48-53.

Gorkom, N.V. 2004. *Abhidhamma and Practice*, http://www.abhidhamma.org/

Gruber, T.A. 1993. A Translation Approach to portable Ontology Specification, *Knowledge Acquisition*, **5**(2) 199-220.

Guarino, N. 1997. Understanding, building and using ontologies, *International Journal of Human-Computer Studies*, **46**, 293-310

Hahn, U. and Romacker, M. 2000. The SynDiCATe text knowledge base generator, *Data and Knowledge Analysis*, **35**(2), 137-159

Hendler, J. and McGuinness, D. L. 2000. The DARPA Agent Markup Language, *IEEE Intelligent Systems*, **15**(6), 67-73.

Horrocks, I., Fensel, D. Broekstra, J., Decker, S., Erdmann, M., Goble, C., Van Hermelen, F., Klein, M., Staab, S., Studer, R. and Motta, E. 2000. *The Ontology Inference layer OIL,* http://www.ontoknowledge.org./oil/TR/oil-long.html.

Heflin, J. and Hendler, J. 2000. Searching the Web with SHOE, *AAII Workshop on AI for Web Search*, WS-00-01, AAAI Press Menlo Park, CA, 2000, 35-40.

Hendler, J. 2001. *Agent and the Semantic Web,* http://www.cs.umd.edu/users/hendler/AgentWeb.

Karp, P.D., Chaudhri, V.K., and Thomere, J. 1999. *XOL: An XML-based Ontology exchange language, Version 3,* http://www.ontoknowledge.org/oil/TR/.

Korhonen, J., Pajunen, L. and Puustjarvi, J. 2002. Using Web Service and Workflow Ontology in Multi-Agent Systems, in: *Workshop on Ontological Modelling for Multi-Agent Systems (EKAW’02).*

Lenat, D. 1995. Cyc: A large-scale investment in knowledge infrastructure, *Communications of the ACM,* **38**(11), 31-40.

Lenat, D., Miller, G. and Yokoi, T. 1995. Cyc, WordNet, and EDR, *Communications of the ACM,* **38**(11), 45-49.

Lassila, O. and Swick, R. 1999. *Resource Description Framework (RDF),* W3C Recommendation, http://www.w3.org/TR/WD-rdf-syntax/.

Lorin, J. M. 1998. *Major Causes of Software Project Failures,* CROSSTALK The Journal of Defense Software Engineering, 09-12

Lu, H. 2004. *Ontology based Agent Services Description and Matchmaking on the World Wide Web,* http://ausweb.scu.edu.au/aw03/papers/lu/paper.html.

Martin, D., Cheyer A., and Moran D. 1999. The Open Agent Architecture; A Framework for Building Distributed Software Systems, *Applied Artificial Intelligence* **13**(1-2), 92-128.

Martin, D., Paolucci, M., Mcllraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Sirinivasan, N. and Sycara, K. 2004. *Bringing Semantics to Web Services: The OWL-S Approach,* Lecture Notes in Computer Science 3387, pp 26-42.

MacGregor R. 1991 The Evolving Technology of Classification-based Knowledge Representation Systems, in: *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, edited by Sowa, J.F., San Mateo, CA: Morgan Kaufmann.

Maedche, A., and Staab, S. 2001. Ontology learning for the semantic web, *IEEE Intelligent Systems*, **16**(2), 72-79.

McEvilley, T. 2002. *The Shape of Ancient Thoughts: Comparative Studies in Greek and Indian Philosophies,* Allworth Press, New York.

McGuinness, D.L. and van Harmelen, F. 2004. *OWL Web Ontology Language Overview,* W3C Recommendation 10th February 2004.

McGuinness, D.L., Fikes, R., Hendler, J. and Stein, L.A. 2002. DAML+OIL: An Ontology Language for the Semantic Web, *IEEE Intelligent Systems*, **17**(5), 72-80.

McGuinness, D.L., Fikes, R., Rice, J. and Wilder, S. 2000. An Environment for Merging and Testing Large Ontologies, in*: Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR 2000),* Breckenridge, Colorado.

Miller, G. 1995. *WordNet: A Lexical database for English*, Communication of the ACM, **38** (11), 39-41.

Mitkas, P.A., Kehagias, D., Symeonidis, A. L., and Athanasiadis, I. N. 2003. A Framework for Constructing Multi-Agent Applications and Training Intelligent Agents, in: *Proceedings of Fourth International Workshop on Agent Oriented Software Engineering*.

Newell, A. 1982. The Knowledge Level, *Artificial Intelligence,* **18**(1), 87-127.

Nwana, H.S. and Ndumu D. T. 1999. Perspectives on Software Agents, *Knowledge Engineering Review*, **14**(2), 1-18.

Nwana, H.S. 1999. Software Agents: An Overview, *The Knowledge Engineering Review,* **11**(3), 205-244.

Nwana, H.S., Ndumu, D.T., Lee, L.C. and Collis, J.C. 1999. ZEUS: A Toolkit for building distributed Multi-Agent Systems, *Applied Artificial Intelligence Journal* **13**(1/2), 129-185.

Nodine, M.H., and Unruh A. 1998. Facilitating Open Communication in Agent Systems: the OnfoSleuth Infrastructure, in: *Intelligent Agents IV: Agent Theories , Architectures and Languages*, edited by Singh, M., Rao, A., and Wooldridge, M., Springer Verlag,, 281-296.

Odell, J., Parunal, H. V.D., and Bauer, B. 2001. Representing Agent Interaction protocol in UML, in: *Agent Oriented software engineering – Proceedings of the First International Workshop AOSE-2000,* edited by Ciancarini, P. and Wooldridge, M.), Springer, Berlin, LACS Volume 1957, pp 121-140.

OntoWeb. 2002. *Ontology-based information exchange for knowledge management and electronic commerce,* IST-2000-29243.

Pan, J. Z. and Horrocks, I. 2002. RDFS (FA) and RDF MT: Two Semantics for RDFS, *Lecture Notes in Computer Science,* Number 2879, pp 30-46, Springer

Paolucci, M., Kawamura, T., Payne, T.R., and Sycara, K. 2002. Semantic matching of Web Services Capabilities, in: *Proceedings of the 1st International Semantic Web Conference (ISWC).*

Paolucci, M., Sirinivasan, N., Payne, Sycara, K., Nishimura, T. 2003. Towards a Semantic Choreography of Web Services: from WSDL to DAML-S, in: *Proceedings of the International Conference on Web Services (ICWS)*

Payne, T.R., Singh, R. and Sycara, K. 2002a. Calendar Agent on the Semantic Web, *IEEE Intelligent Systems*, 84-86.

Payne, T.R., Paolucci, M., Singh, R. and Sycara, K. 2002b. Communication Agents in Open Multi-Agent Systems, in: *Proceedings of First GSFC/JPL Workshop on Radical Agent Concepts (WRAC),* McLean, VA, USA, 365-371.

Pinto, H.S., Peralta, D.N. and Mamede, N. J. 2002. Using Protégé in Reuse Processes, in: *Proceedings of the OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002,* 15-26.

Quine, W. O. 1961. From *a Logical Point of view. Nine Logico-Philosophical Essay,* Cambridge, MA: Harvard University Press.

Rao, A.S. and Georgeff, M.P. 1991. Modelling rational Agents within a BDI-Architecture, in: *proceeding of the Second International Conference on Principles of Knowledge Representation,* 473-484.

Reed, S. L. and Lenat, D. B. 2002. Mapping Ontologies into Cyc, in: *proceedings of AAAI 2002 conference workshop on Ontologies for the Semantic Web,* Edmonton, Canada, 2002

Ricordel, P.M. and Demazeau, Y. 2000. From Analysis to Deployment: a Multi-Agent Platform Survey, in: *Proceedings of ESAW00,* http://lia.deis.unibo.it/confs/ESAW00/.

Russell, S. & Norving, P. 2003. *Artificial Intelligence: A Modern Approach,* Prentice Hall.

Rzevski, G. Skobelev, P. 2004. *Magenta Multi-Agent Technology: Magenta Platform Version 2*, Whitepaper, Magenta Corporation Ltd.

Rzevski, G. 2003. On the Internet-Based Global Economy. MADIRA Paper 002, Version 1.1 published on MADIRA website www.brunel.ac.uk/research/madira

Rzevski, G. 2004. On Multi-Agent Systems and Distributed Intelligence. MADIRA Paper 021, Version 2.1, published on MADIRA website www.brunel.ac.uk/research/madira.

Rzevski, G.2005. Ontology and Relativity. Volga Conference on Complex Adaptive Systems, Samara, Russia.

Shamsfard, M. and Barforoush, A.A. 2004. Learning ontologies from natural language texts, *International Journal of Human-Computer Studies* **60**, 17-63.

Silverman, A. 2003. *Plato's Middle period metaphysics and Epistemology*, The Stanford Encyclopaedia of Philosophy (Summer 2003 Edition), edited by Edward N. Z.

Siegel. N., Goolsbey, K., Kahlert, R. and Matthews G. 2004. The Cyc System: Notes on Architecture, http://www.cyc.com, Cycorp, Inc, 2004

Sitagu Sayadaw U Nyanissara (2004), *Abhidhamma and Vipassana*, http://www.triplegem.plus.com/dhamaj2.html.

Spackman, K.   2000. SMOMED RT: *Systematized Nomenclature of Medicine*, Reference Terminology, College of American Pathologists, Northfield, IL.

Spiby, P. 1991. *EXPRESS Language Reference Manual*, International Standard ISO 10303-11, Project Three ("Express Language") Working Group Five ("STEP Development Methods") of Subcommittee Four ("Manufacturing languages and Data") of ISO Technical Committee 184 ("Industrial Automated Systems and Integration"); International Standards Organization.

Sowa, J. F. 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundation*, Brooks Cole Publishing Co., Pacific Grove, CA.

Sowa, J. F. 2001a. *Building, sharing and Merging Ontologies*, http://www.jfsowa.com/Ontology/ontoshar.html.

Sowa, J. F. 2001b. *Application of Ontology*, http://www.jfsowa.com/talks/Ontology.html.

Sowa, J.F. 1995. Top-Level Ontological categories, *International Journal of Human-Computer Studies* **43**, 669-685.

Stockheim, T., Nimis, J., Scholz, T. and Stehli, M. 2004. How to Build a Multi-Multi-Agent System – The Agent.Enterprise Approach, in: *proceedings of Sixth International Conference on Enterprise Information Systems (ICEIS 2004)*, Porto, Portugal.

Sycara, K., Klusch, M., Widoff, S. and Lu, J. 1999. *Dynamic Service Matchmaking Among Agents in Open Information Environments*, SIGMOD Record (ACM Special Interest Group on Management of Data), **28**(1), 47-53.

Sycara, K., Decker, K., and Willianson, M. 1997, Middle-Agents for the Internet, in: *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI).*

Tamma, V., Wooldridge, M. and Dickinson, I. 2002, An Ontology for automated negotiation, in: *Proceedings of the Second International Workshop on Ontologies in Agent Systems, 1st International Joint Conference on Autonomous Agent and Multi-Agent Systems*, 62-69.

Tim Berners-Lee, Hendler, J. and Ora Lassila 2001. The Semantic Web, *Scientific American.*

Warren, D.H.D. and Pereira, F.C.N., 1982. An efficient easily adaptable system for interpreting natural language queries, *Computational Linguistics* **8** (3-4), 110-122.

Wooldridge, M. 2000. *Reasoning about Rational Agents*, The MIT Press, Cambridge MA

Wooldridge, M. 2002. *Introduction to Multi-Agent Systems*, John Wiley & Sons Ltd.

Wooldridge, M. 2004, *Responding to Real-World Complexity: Introduction to Multi-Agent Technology – Magenta Whitepaper*, Magenta Technology.

Wooldridge, M., Jennings, N. R. and Kinny, D. 2000. The Gaia methodology for Agent-Oriented Analysis and Design, *Journal of Autonomous Agents and Multi-Agent Systems*, **3**(3), 285-312.

Yokoi, T. (1995), The EDR electronics dictionary, *Communications of the ACM*, **38**(11), 42-44.