ITcon
www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

TOWARDS AGENT-AUGMENTED ONTOLOGIES FOR EDUCATIONAL VDC APPLICATIONS

PUBLISHED: June 2010 at http://www.itcon.org/2010/24

EDITOR: Issa R R A

Esther Obonyo

Rinker School of Building Construction University of Florida, Gainesville, Florida, USA.

obonyo@ufl.edu

SUMMARY: *Existing Virtual Design and Construction (VDC) applications still lack an overall integration scheme for addressing the challenges inherent in knowledge management. Resolving these challenges would greatly increase the benefits accruing from educational initiatives using such applications. This paper discusses the potential for addressing these challenges using Semantic Web technologies. More specifically, it focuses on the feasibility of implementing agent-augmented ontologies. Despite the significant amount of effort that has been invested in building domain ontologies for virtual design and construction applications, leading researchers in this area have acknowledged that much more research and development work is needed. A primary area of need is "the development of appropriate mechanisms and tools for information extraction and document annotation." Such a mechanism would reduce the complexities inherent in developing domain ontologies. The main aim of this paper is to discuss how this desired mechanism can be realized through leveraging on the agent paradigm.*

KEY WORDS: *Software Agents, Ontologies, Knowledge Management, Virtual Design and Construction.*

REFERENCE: *Obonyo E (2010) Towards agent-augmented ontologies for educational VDC applications, Special Issue Advanced Digital Technologies for Built Environment Education and Learning, Journal of Information Technology in Construction (ITcon), Vol. 15, pg. 318-334 , http://www.itcon.org/2010/24*

1. INTRODUCTION

In recent years, there has been a rapid growth in the use of VDC applications for design and functionality analysis in the early stages of the project development cycle (Kunz and Fischer, 2009). The motivation for using the resulting virtual mock-ups is based on the need to increase constructability during the early phases of a project. The virtual mock-ups offer the project team a way of ensuring that any prefabricated components have been designed for assembly. Some very specific benefits of using such models include (Ding et al. 2003): 1) Improving the visualization of the building model; 2) Representation of multi-dimension design space through supporting the addition of new components or linking with various application packages; 3) Providing real-time interactions among the design team, thus enabling the exploration of alternative ideas and design plans produced in a real design process; 4) Providing multi-user real-time collaboration for problem-solving, and; 5) Linking with a broad range of information and accessing various databases or different domain application models through a network.

Although significant strides have been made in refining the functionality of VDC applications, there is still no overall integration scheme that addresses the challenges inherent in knowledge management. The fundamental knowledge management challenge being addressed by several VDC researchers is similar to what was described in the GENIAL project as the "digital anarchy" (Radeke, 1999). This was their characterization of incompatible Construction IT applications that cannot readily inter-operate or build upon each other. Such incompatibilities limit the benefits accrued from using an integrated model, especially given that a significant part of the contextual issues is often in the form of tacit knowledge. Standardization has its benefits; but on their own standards cannot adequately address this problem.

VDC applications are increasingly being used to support learning and educational activities. Examples include the initiatives reported by Muramoto et al. (2008) and Issa et al. (2008). Such applications have inherited the digital anarchy problem, and within an educational setting it becomes more critical as the targeted end users are novices in construction who can greatly benefit from context-specific information that may be difficult to codify for inclusion in a building model. This paper discusses how strategies leveraging on emerging Semantic Web technologies can be used to enhance the functionality of virtual models through addressing key information integration issues. There is a specific focus on the potential for using agent-augmented ontologies to enhance knowledge sharing in VDC applications. Section 2 provides background information on Semantic Web technologies. This is followed by an overview of related work. The methodology for the research (an agent-based approach) is described in Section 4. This is followed by a description of the implemented proof of concept. The paper ends with a discussion giving an overview of the main conclusions. It also identifies the main limitations in the approach as well as future directions for the research.

2. CONTEXTUAL BACKGROUND

Ideally, virtual design and construction applications should encapsulate the views of participants from different domains using specialized software applications including the various CAD packages, estimating and scheduling software, project planning software and Enterprise Resource Planning (ERP) systems. Because of the reliance on historical data from previous jobs, it may be necessary to include a mechanism for incorporating data from several legacy applications. In addition to these challenges, projects typically involve several participants from conceptual design to detail design, manufacturing of prefabricated components, executing job-site operations, commissioning and using the facility, and decommissioning it at the end. This creates a need for dynamism in any system based on information representing the full life cycle of any constructed facility.

This type of dynamism is actually articulated in Berners-Lee's (1989) vision for the Web. A number of researchers affiliated to the W3C are keen to advance the present functionality of the Web to a level where it will hold direct links to the information contained in the documents displayed on the Web (Berners-Lee et al. 2001). The W3C's Web of the future, which will hold machine-processable information, has been very broadly defined as the Semantic Web. It will allow people to find, share and combine information more easily (Hendler et al. 2002). W3C has set out to define and link the Web in a way that enables it to be used for more effective information discovery, automation, integration and re-use across applications. Hendler et al. (2002) identified the four key enabling technologies for the Semantic Web as: 1) XML – adds arbitrary structure; 2) RDF – provides a common framework for representing metadata across many applications; 3) Ontologies – store formal definitions of relations among terms, and; 4)Software agents – automate tasks.

The research described in this paper leverages on AEC-specific standardization efforts that have extensible developed schemas created initially using IFCs and more recently using XML and RDF (examples of which have been identified elsewhere in this paper). The discussion in the subsequent paragraphs focuses on advancing the use of ontologies using the agent paradigm. Both terms lack formal or universally accepted definitions. The most frequently quoted definition for ontologies is Gruber's (1994) description of ontology as "a formal definition of conceptualization." Other commonly cited definitions include the ones provided by McGuinness (2003) and Gomez-Perez and Corcho (2002). For an in-depth critique of existing definitions, see Guisheng and Qiuyan (2008). They extended Gruber's definition of ontologies into a structure: <*O, L*>. '*O'* here is an ontology structure playing the role of an explicit specification of a conceptualization of some domain, while '*L'* is a corresponding lexicon, providing the agreed vocabulary for communicating the former. Ontologies are therefore semantic-rich metadata that can be used to obtain relative information about knowledge bases at the lower layers (Yu and Zhiping 2009). In simple terms, ontologies provide “a way of representing common understanding of a domain” (Sanchez et al. 2007). They are increasingly being used to address some of the challenges in sharing knowledge (Yu and Zhiping 2009).

The other pivotal technology in this research is the notion of software or intelligent agents. It is not easy to define the term ‘agents’. Nwana (1996) provides a number of explanations for this difficulty: it is a common term in everyday conversion; it encompasses a broad area; it is a meta-term, and researchers in this area have come up with such synonyms as ‘bots’, ‘spiders’ and ‘crawlers’. Software agents in this project have been explored from the viewpoint of leading researchers. Brustoloni (1991), Ferber (1999), Foundation for Intelligent Physical Agents (FIPA), Architecture Board (2001), Jennings and Wooldridge (1998), Lieberman (1997) and Maes (1994) have all defined the term ‘agent’ in various ways based on their research interests. It is, however, possible to extract some common attributes of agents from these definitions. There is a general consensus that software agents exist in an environment. They can sense the conditions in the environment, and such senses may affect how they act in the future. Software agents are also perceived as adaptive components that are capable of learning. They are proactive, exhibiting goal-directed behaviour. The execution of tasks occurs autonomously (without human intervention). Based on an analysis of these attributes, the term ‘agents’, as used in this paper, very loosely refers to systems capable of autonomous, purposeful action in the real world.

3. RELATED WORK

A previous section identified the GENIAL as an example of an initiative directed at addressing the digital anarchy problem. The objective of the GENIAL project was to define an open architecture and to establish a common semantic infrastructure (Radeke, 1999). The solution adopted was largely based on the use of standards. The output of the GENIAL project was extended in the eConstruct project. The focus of the eConstruct project was to develop an XML vocabulary (bcXML) for the European building and construction industry (Stephens et al. 2002). A second project that emerged from the GENIAL project was a collaborative project between Taylor Woodrow and Loughborough University, whose objective was to extend the discrete product search in GENIAL into a product schedule that could be used to perform product comparisons across different suppliers (Ugwu et al. 2002). This work was further extended in a project focusing on the specification and procurements of construction products (Obonyo et al. 2004 and 2005). Parallel efforts in the e-bip project also tried to address the digital ‘anarchy’. These resulted in a business-to-business broker service for Small and Medium-sized Enterprises (SMEs) in the construction tile supply chain (Thiels et al. 2002).

The digital anarchy problem is not unique to the construction industry, and the potential solutions being investigated are based on precedence from other fields. The use of ontologies to address knowledge-sharing issues is exemplified in efforts directed at ensuring consistency and correctness by formulating constraints on the content of information, and also to create libraries of interchangeable and reusable models. Examples include work reported by Gomez et al. (2001), which demonstrated how ontologies can be used to support inference for deriving additional knowledge from a set of facts. Recent examples of ontology-based knowledge management applications have been reported by Kim et al. (2005), Assali et al. (2007), Sun et al. (2008), Liu et al. (2008), Yu and Zhiping (2009) and Li et al. (2009). Kim et al. (2007) investigated the deployment of ontologies within a virtual prototyping setting. Assali et al. (2007) developed an ontology-based knowledge management system for indexing and retrieving information about a domain-specific corpus of resources for industrial safety. Sun et al. (2008) integrated the use of ontologies with knowledge management, resulting in a product knowledge model that could be used by stakeholders in the product development process to locate and re-use information. Liu et al. (2008) demonstrated the use of ontologies in the development of a semantic retrieval system for high-speed railways. Yu and Zhiping (2009) proposed a model of ontology-based Knowledge Management (KM) for Intelligent Tutoring Systems. Li et al. (2009) proposed a framework for using ontologies in KM.

Examples of recent work in construction-specific ontologies include the development of a taxonomy for construction concepts by El-Diraby et al. (2005), Issa and Mutis’s (2006) definition of a framework for semantic reconciliation, and Pan’s (2006) Semantic Web-based information management framework. More recently, Liao et al. (2009) presented ontologies as a possible way of addressing the challenges of information flow within the context of mobile knowledge service during the inspection of construction projects. This notwithstanding, much more works needs to be done before the use of ontologies for addressing knowledge-sharing needs can become a widespread phenomenon. There is a specific call for "the development of appropriate mechanisms and tools for information extraction and document annotation" (Anumba et al. 2008). This is a role that can be played by agents as explained in the subsequent section.

4. METHODOLOGY: AN AGENT-AUGMENTED APPROACH

Software agents have a large repertory of attributes, as indicated in the preceding section, which could potentially extend the performance of business information systems developed using conventional techniques such as object-oriented programming (Fingar, 1998). The key features in a software agent's basic anatomy include: autonomy, reactivity and the ability to communicate, plan and set goals, reason about actions, and learn. Agents can therefore serve as the basis for developing software solutions that can effectively automate and augment business processes. The precedence for using an agent-based approach as presented in this paper is derived from the diverse experiences of leading agent technology researchers including Ferber (1999), Kashyap (1997), Parunak (1996), O'Malley and DeLoach (2001), Bielawski, L and Lewand (1991and Jennings (1999).

A key argument for using agents in this context is that they simplify communication. It was essential to establish that the requirements of KM for VDC applications could be bundled into manageable agent tasks that would not take an individual human being more than a few hours to solve; otherwise writing rules for automation becomes a tedious and complicated task. The greatest value from using software agents can be realised only when it is possible to precisely articulate the problem. The benefit of using agents here accrues from the time saved through automating the repetitive actions inherent in the execution of the tasks. The agent model in Listing 1 describes a formwork connection specification agent. This example illustrates how the various aspects of KM can be readily broken down into simple agent tasks. It is evident that virtual design and construction of components as an application domain is narrow, specific and restricted enough to allow for the easy deployment of agents.

As O'Malley and Scott (2001) pointed out, an ideal application domain for the deployment of agents must demonstrate a need for the advanced level of autonomy inherent in agent-based systems. The deployment of agents in a given domain makes sense only when there is a need for a system that is capable of acting on behalf of a user. Several knowledge-sharing problems can be addressed through the generation of ontologies for the construction domain. However, the generation of ontologies constitutes executing several time-consuming tasks. The case for using agents here is in direct response to Anumba et al.'s (2008) call for improving the development of ontologies through the use of appropriate mechanisms and tools for information extraction and document annotation.

LISTING 1: Agent Model (Internal)

Scenario Outline: A buyer's agent is required to match the designer's requirements and the manufacturers' brochures. The building codes govern the choice, cost and type of product the concrete subcontractor is allowed to use in the forming system. Here we consider an agent, within a formwork engineer's office, to which the engineer can give product requirements. The agent then explores the various product possibilities that comply with design standards. The user interface agent asks the designer to decide which options to pick, generating a detailed specification for the product. To perform comparisons across different manufacturers' brochures, the operation is executed within the context of a multi-agent community, which has specialist agents responsible for knowledge mapping to ensure the seamless flow of information.

Formwork Connection Agents
Identity: Perform comparisons across the products described in different manufacturers' brochures.
Roles: Extract connection requirements and structure specification information using preferred schema, and identify suitable options.
Interactions with environment:
Sensory input: The formwork engineer initiates the process by selecting a desired category of attributes and specifies desired values for the selected attributes, for example, type of connection (wood screws, nails, spikes, drift bolts, drift pins), diameter of fastener (1/2 inch), withdrawal design values – Z parallel (2400lbs) and edge distance requirements (0.5 inch).

Acquaintances: In this scenario, the tasks are executed within the context of a multi-agent community, which has specialist agents responsible for knowledge mapping to ensure the seamless flow of information from different brochures.

Resource ownership and access: The formwork engineer has read-only access to the manufacturer's valves brochures to enable the electronic viewing of product attributes.

Actions:
(Communicate) Get valve attribute category from the designer, for example, type of connection, diameter of fastener, withdrawal design values and edge distance requirements.
(External) Gather information on options from manufacturers' brochures.
(Internal) Evaluate the options and decide on the best selection based on the specified constraints.
(Communicate) Suggest options to the formwork engineer.
(Communicate) Get choice from the formwork engineer.
(Communicate) Get choice from the designer.
(Internal) Construct detailed specification for the product and store in knowledge base for future reference.

Mental state and behaviour:
Purpose: The agent's goal is to extract performance requirements from an engineer and turn this into detailed product specification by suggesting alternatives to the user based on the constraints imposed by the engineer and product availability as documented in the manufacturer's brochure. Once the engineer makes the choice, the agent generates a detailed product specification that can be used for bidding.
Behaviour: The connection agent is not able to confirm the specified fasteners without the formwork engineer first confirming a choice.

Knowledge and beliefs:
- How to generate specifications based on an engineer's requirements.
- How to access and use information from a manufacturer's brochure.

Agents generally exist in the context of multi-agent communities and their behaviour derives from the interaction among constituent agents (Huhns 2000). Problems that require solutions that can handle concurrent execution of multiple tasks with conflicting goals are ideal candidates for agent-based solutions. Other qualifying features include the existence of multiple stakeholders with different interests, and the existence of problems that are both too complex for a monolithic solution and too difficult to decompose into independent sub-problems. Based on the potentially large number of stakeholders involved in design and construction activities, KM for VDC is an ideal environment for deploying a community of agents.

The approach adopted in this paper is based on the premise that an ideal agent-oriented methodology captures all the elements necessary for the implementation of a multi-agent community. An agent-based model comprises a set of agents, a set of agent relationships and a framework for simulating agents' behaviours and interactions (North and Macal 2007).

Agent-to-agent interactions: In a multi-agent environment, the problem is fragmented into interdependent sub-problems (Jennings et al 1998). Agents are not mutually exclusive, and agent interdependency can take the form of overlapping sub-problems. The desired solution satisfies the overlapping constraints of the sub-problems. Agents have to make decisions such as designating agents to execute specified tasks at a given time and communicate their results. It is necessary to coordinate the decisions of agents in the sub-problems in order to achieve an optimal overall solution. The agents' needs determine not just the nature of sub-problem interdependency, but also the other agents' current state of problem-solving and the status of network resources. Agents develop a unified plan by recognizing and avoiding or resolving sub-goal interactions.

Inter-agent communication: Communication enables agents to cooperate, coordinate their actions and carry out tasks. Without communication, they would be "isolated, individual, deaf and dumb to other agents" (Feber 1999). The main goal here is developing a common agent communication language, a protocol and a format for the content, as well as a mode of transport (Sankhwal 2000). Many platforms leverage the FIPA Agent Communication specifications, which deal with Agent Communication Language (ACL) messages, message exchange interaction protocols, speech act theory-based communicative acts and content language representations (FIPA 2001).

Coordination in a multi-agent system: Coordination involves various approaches, including synchronization of, actions, coordination through planning, reactive coordination and eco-problem-solving (Ferber 1999). By synchronizing actions and access to resources, it is possible to devise a mechanism that allows actions to be articulated correctly. Synchronization provides coherence and prevents interference. Coordination by planning extends artificial intelligence planning paradigms adopted in single-agent systems to a community of agents. In reactive co-ordination, the agents' actions are responses to their perception of the environment. Spatial relationships define constraints and capacities for actions.. In eco-problem-solving, the problem is reformulated into a set of interacting agents pursuing individual goals. It consists of a core specifying that the protocol is to be followed by agents and a module encoded with the individual agent's behaviour, which is moulded by the application domain.

Negotiation: Negotiation resolves disparities in a multi-agent system (Sycara 1998). In a community of agents, each will be self-interested. The need for negotiation is created by the existence of a conflict requiring a resolution among a set of self-interested agents under conditions of bounded rationality and incomplete information. The system needs to enable automated negotiations and coalition formation. Automated negotiation can be structured into protocols and strategies.

A robust deployment framework would, at the very least, support the implementation of systems that exhibit the key features of agent-based systems as discussed in the preceding paragraphs. Based on the findings of previous efforts (Obonyo et al. 2004 and 2005), JADE was chosen as the Agent Platform. JADE is based on the FIPA specifications. These specifications enable interoperability within and across agent-based applications. The core components of a FIPA-compliant Agent Platform are the Directory Facilitator (DF), Agent Management System (AMS), Agent Communication Channel (ACC) and Internal Platform Message Transport (IPMT). The core components have been depicted in Figure 1. The DF and AMS provide "yellow pages" and "white services", respectively, to other agents. The ACC supports inter-agent communication. The ACC supports interoperability while the IPMT provides a message-forwarding service for agents on a particular platform (O'Brien and Nicol 1998).

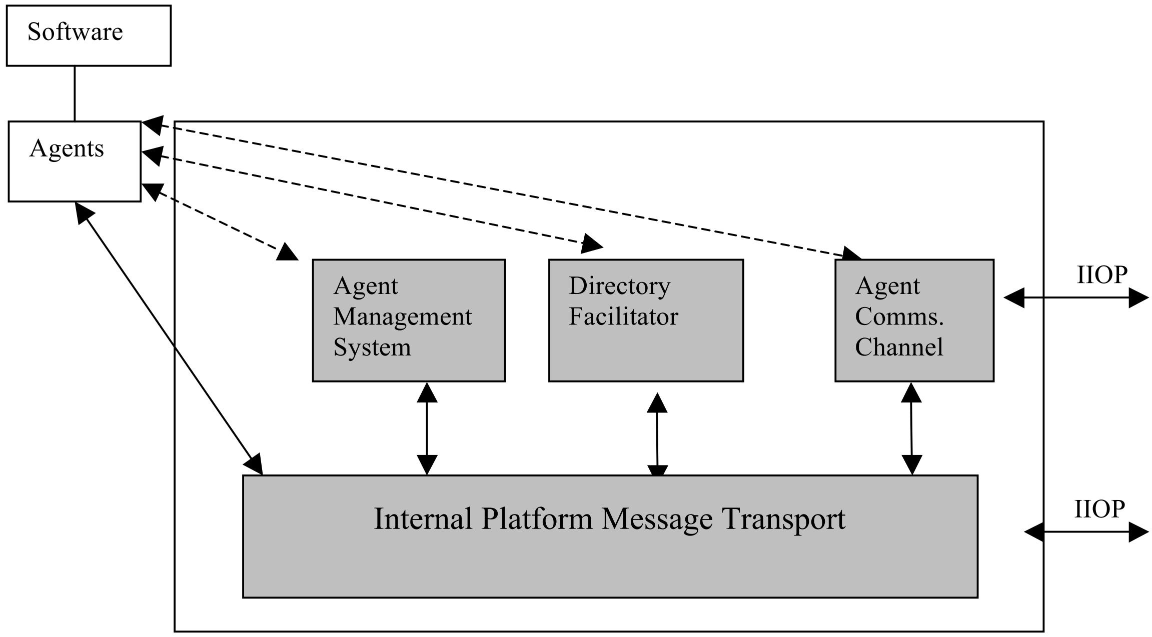

FIG 1: FIPA-OS Agent Platform
Source: Poslad et al. 2000

5. PROOF OF CONCEPT: IMPLEMENTING AGENT-BASED WRAPPERS FOR ONTOLOGIES

It is important to note that the intent here was to demonstrate the potential for wrapping ontologies using an agent-based approach to improve the seamless flow of information. This would also promote the re-use of existing ontologies. There are examples of ontologies being developed for several industrial applications. Given the relatively slow rate at which construction industry-specific ontologies are being developed, there is a need to explore the opportunities for leveraging on the ones which have already been developed, many of which are available in the public domain as free resources. A focused search targeting ontologies that can be edited for re-use identified several examples, including the ones shown in Table 1. These existing ontologies essentially offer a framework for representing information for architecture, engineering, construction and facility management projects. These descriptions deal not only with full 3D geometry, but also relationships, process, material, cost and other behaviour data.

TABLE 1: Potential for Reusing Existing Ontologies

Existing Ontologies	**Exemplary use in Construction**	**Web Source**
Ontology for 3D Semantic Virtual Environments	VDC Applications	http://www.uv.es/~agentes/SVECore.owl
Wood Ontology	Structural Lumber for Formwork	http://abulaish.com/Wood_Ontology/woodontology.xml
Infrastructure Products	Water, Wastewater, Electricity	http://individual.utoronto.ca/hesham/DetailedResearch.htm
Performance	Process Improvement	http://swap.uib.es/ontologies/performance.owl
Context	Context-aware KM	http://on.cs.unibas.ch/owl/1.0/Context.owl
Environmental Sustainability	Sustainability Assessment	http://sweet.jpl.nasa.gov/2.0/envirSustainability.owl
Product Design Ontology	Designing Prefabricated Elements	http://www.aimatshape.net/resources/aas-ontologies/pdontology.owl
Virtual Human Ontology	Managing Labour	http://www.aimatshape.net/resources/aas-ontologies/virtualhumansontology.owl/view
Engineering Design Ontology	Designing Structural Elements	http://edesign.ecs.umass.edu/ontologies/Framework2.0/Design_Model2.0.owl
Shuttle Crew Ontology	Managing Labour	http://protege.cim3.net/file/pub/ontologies/shuttle/shuttle-crew-ont.owl

The building of ontologies, and more specifically, harvesting ontologies from the Web, can be supported using a multi-agent approach (Wooldridge 2002). This is also based on precedence – information agents have been used to retrieve information from the Web (Maes 1994).

Such agents use learning and classification techniques to harvest content through identifying information from distributed knowledge bases and extracting relevant concepts. The use of such agents has been validated by the author in a previous project for focusing on the specification and procurement of construction products (Obonyo et al. 2004 and 2005). This section focuses on the use of agents to dynamically populate an ontological knowledge base after information-gathering agents have extracted the relevant information from the Web. A proof of concept was implemented to illustrate the coupling of agents with ontologies. The information used in the proof of concept was based on managing the flow of information that is required for the assembly of prefabricated formwork components (Figure 2).

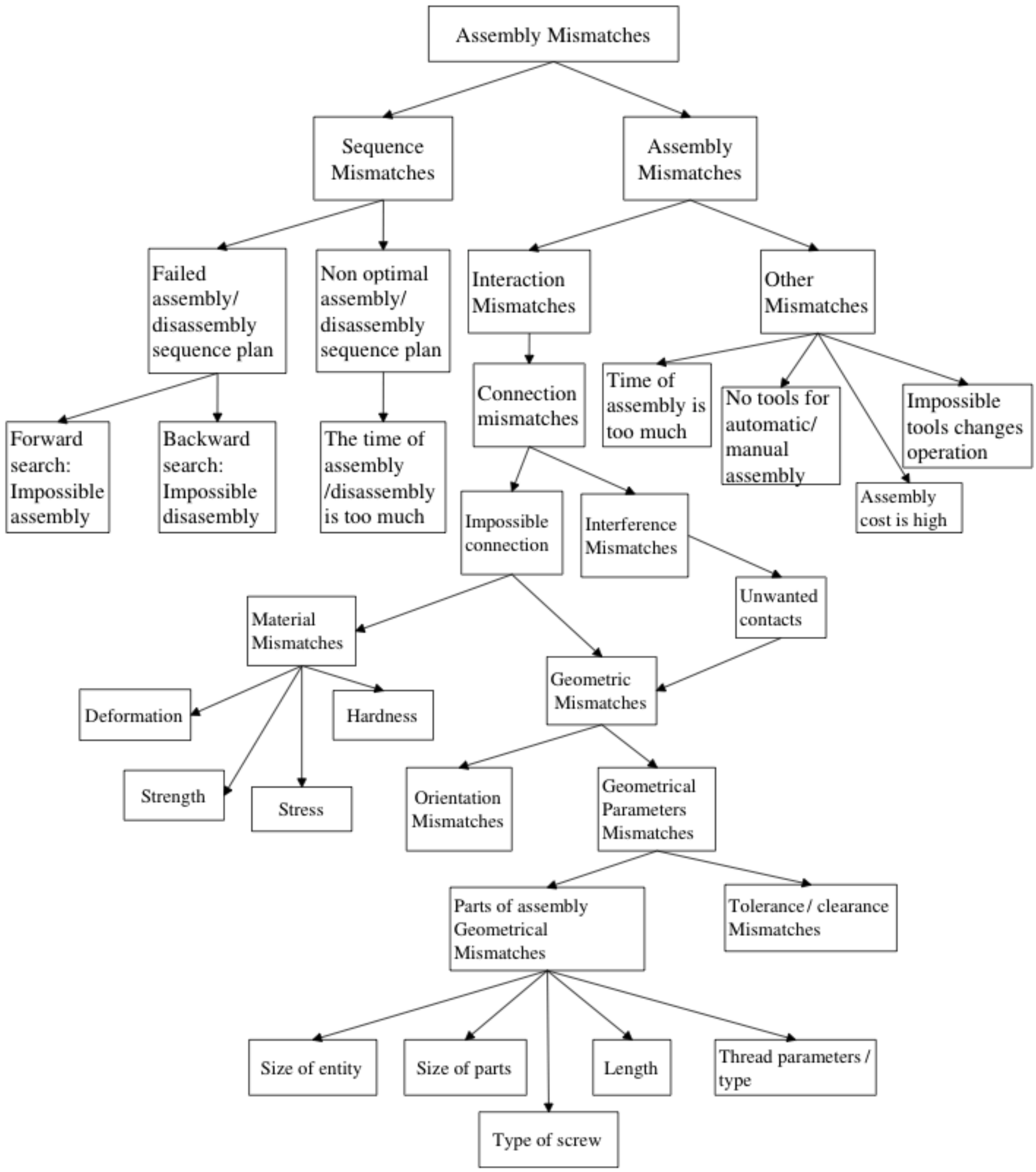

FIG 2: A Taxonomy for Parts of Assembly
Adapted from Taratoukhine and Bechkoum (2000)

In the last decade several tools for building ontologies have emerged (AI^3 2010). Protégé **(**URL1**)**, a flexible, java-based, open-source ontology editor, was selected for use in this example (Figure 3). This was largely due to the existence of an open-source plug-in that allows the transformation of ontologies implemented in protégé to be exported into java objects that can be evoked by JADE agents. An ontology for parts of assembly for prefabricated formwork components was implemented by extending the class ontology predefined in JADE and adding a set of element schemas describing the structure of concepts, actions, and predicates which comprise the content of agent messages. In a conventional object-oriented approach, java objects are used directly in the content of messages. Through the agent-oriented approach, the objects are wrapped using specific terms and concepts. This is presented here as a key advantage of the agent-oriented approach.

Ontologies can be wrapped using a common superclass, which in this case is the **ObjectSchema** class within JADE. There are two approaches to doing this: 1) using the **AgentActionSchema** class which inherits from the **ConceptSchema** class, which is, in turn, a subclass of the **TermSchema** class, or 2) using the **PredicateSchema** class which inherits from the **ContentElementSchema** class. The use of the **PredicateSchema** and **AgentActionSchema** ontology objects is governed by FIPA specifications.

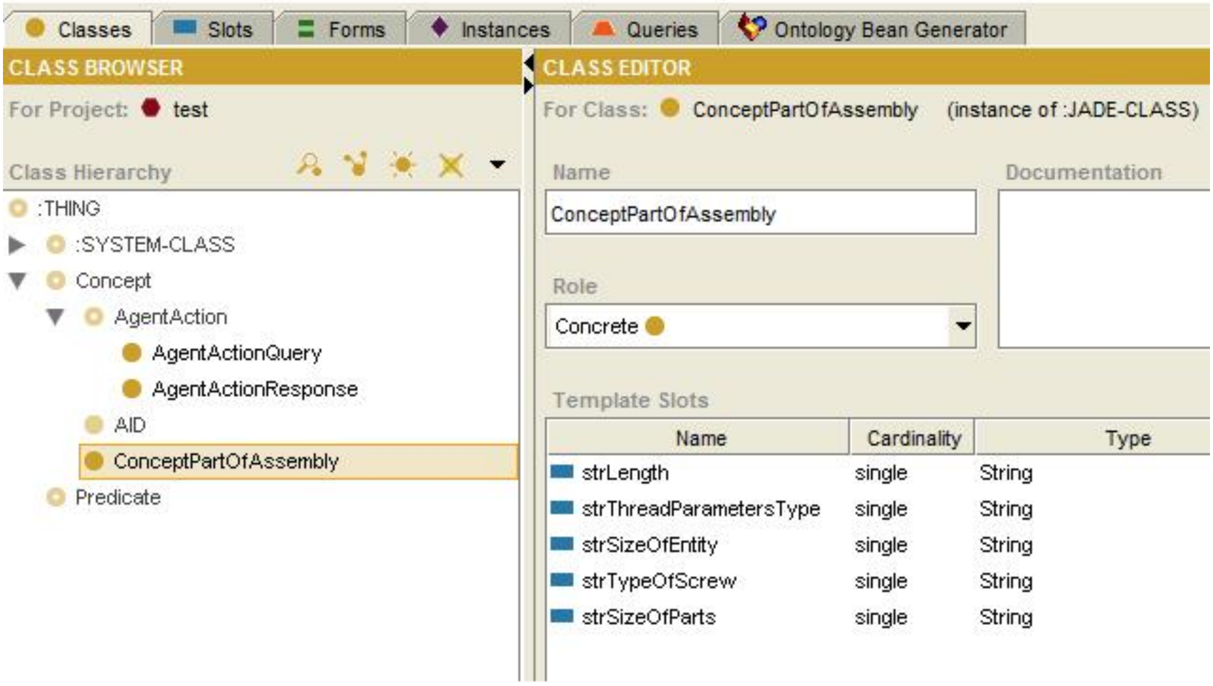

FIG 3: A Screenshot of the Ontology Implemented in Protégé

When an agent is seeking information on whether or not a given preposition is true, it is defined in JADE by a java object implementing the interface *Predicate*. When an agent is requesting another agent to perform a specific task, then the content of the message constitutes an action and is therefore implemented by a java object implementing the *AgentAction* interface. Objects that are neither propositions nor agent actions are defined as concepts. This information is exported to Java objects using a plug-in developed specifically for JADE agents (see Listing 2).

LISTING 2: Extract of the Code for the Java Objects

```
public class AgentActionQuery implements AgentAction {

   private AID aidBuyer;
   public void setAidBuyer(AID value) {
    this.aidBuyer=value;
   }
   public AID getAidBuyer() {
     return this.aidBuyer;
   }

public class ConceptPartOfAssembly implements Concept {
   private String strLength;
   public void setStrLength(String value) {
    this.strLength=value;
   }
   public String getStrLength() {
     return this.strLength;
   }

   private String strThreadParametersType;
   public void setStrThreadParametersType(String value) {
    this.strThreadParametersType=value;
   }
   public String getStrThreadParametersType() {
     return this.strThreadParametersType;
   }
public class PartsOfAssemblyOntology extends jade.content.onto.Ontology  {
  //NAME
  public static final String ONTOLOGY_NAME = "PartsOfAssembly";
  // The singleton instance of this ontology
  private static ReflectiveIntrospector introspect = new ReflectiveIntrospector();
  private static Ontology theInstance = new PartsOfAssemblyOntology();
  public static Ontology getInstance() {
     return theInstance;
  }
```

In this example, no specific values were assigned to the ontology created in Protégé. Specific values are assigned through agent-based communication within JADE to enable the dynamic generation of information. The JADE agents' communication is based on FIPA-Semantic Language, which enable interoperability within and across agent-based applications. In the proof of concept there are the seller and buyer agents, representing the parts trader and the buyer or the end user respectively. Listing 3 is an extract of the agent conversation protocol for this scenario.

LISTING 3: An Extract of the Agent Code

```
public class AgentPartSeller extends Agent
{
  private ContentManager manager = (ContentManager)
getContentManager();
   // This agent "speaks" the SL language
   private Codec codec = new SLCodec();
   // This agent "knows" the  ontology
   private Ontology ontology =
PartsOfAssemblyOntology.getInstance();

   protected void setup()
	{
		manager.registerLanguage(codec);
		manager.registerOntology(ontology);

		registerPartSellingService();
		addBehaviour(new
BehaviourReceiveQueryFromBuyer(this));
		//addBehaviour(new
BehaviourReceiveResponse(this));
   }

	public void registerPartSellingService()
	{
		// Register the part-selling service in the yellow
pages
		DFAgentDescription dfd = new
DFAgentDescription();
		dfd.setName(getAID());
		ServiceDescription sd = new ServiceDescription();
		sd.setType("part-selling");
		sd.setName("JADE-part-trading");
		dfd.addServices(sd);
	}
```

The wrapper around the ontology enables the schema for the domain to be populated and queried dynamically by the agents representing the different stakeholders. For proof of concept, the parts trader's agent accesses the information manual and populates the schema with specific instances for different attributes of connections. The buyer's agent checks the end user's specification against the knowledge base to ensure that there is no mismatch. This exchange has been depicted in Figure 4.

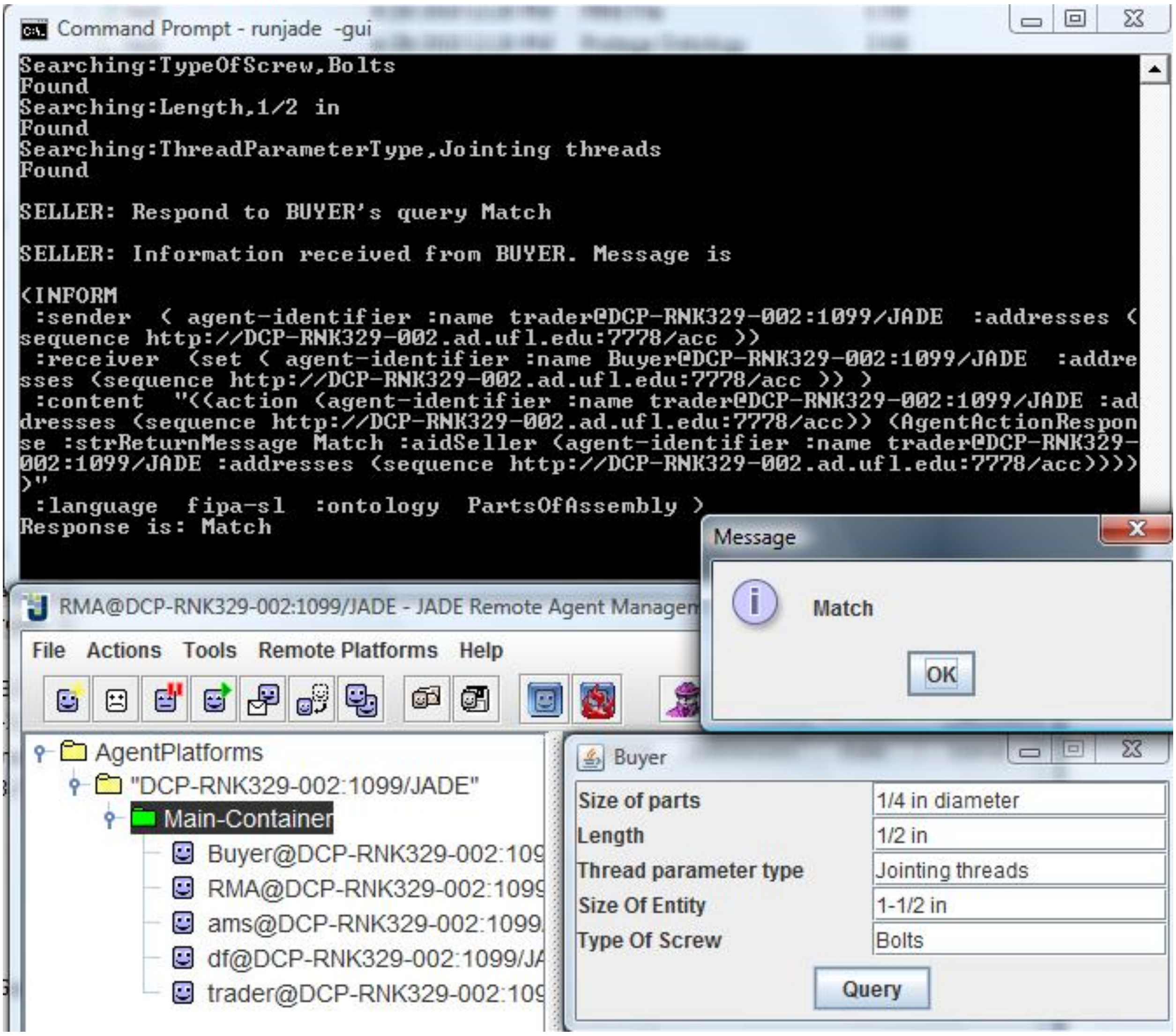

FIG 4: Agents Populating and Querying Parts of Assembly Ontology

The implemented proof of concept demonstrates the feasibility of implementing an agent-augmented, ontology-based approach for KM. The proposed approach would significantly advance the rate at which construction industry-specific domain ontologies are being deployed through, for example, having wrappers integrating ontologies developed by different people. Given that communication between the agents in the system is based on the FIPA specification language, the approach would greatly advance the efforts directed at addressing interoperability challenges. This would in turn advance the efforts directed at realising the seamless flow of information within VDC applications.

6. DISCUSSION AND CONCLUSION

Although significant strides have been made in refining the functionality of VDC applications, there is still no overall integration scheme that addresses the challenges inherent in knowledge management. There are several initiatives directed at leveraging on Semantic Web technologies, to address these challenges. Many of these efforts are directed at the definition and development of domain ontologies, which is a very information-intensive activity. Such efforts progress at a relatively slow pace, largely because of the sheer volume of unstructured information or semi-information that has to be accessed. In addition, there are duplicated ontologies developed by different people that need to be mapped onto each other for the seamless flow of information.

Since the 1990s, there has been significant work that has gone into using software agents to automate some of the tasks inherent in information retrieval. As demonstrated through the proof of concept, there is latent potential for using agent-based technology to enhance the implementation of dynamic domain specific ontologies. This would advance VDC applications through enabling the development of software components that automate information extraction and document annotations. Through this approach, end users can also easily track the logic behind decisions being made during design and construction processes. From an educational perspective, this would be advantageous given the growing demands for educational institutions to graduate students who, in addition to being technically savvy, are also creative thinkers. The type of information that would significantly shift classroom focus from the traditional disciplinary lines is often one that constitutes soft issues that do not lend themselves readily to coding or presentation in a CAD model. This information can be made available within a context of an agent-augmented, ontology-based KM framework.

This notwithstanding, it is important to bear in mind that agent technology is still relatively new. Large-scale deployment of the approach proposed in this paper would therefore be hampered by inadequacies in the existing agent infrastructure (Luck et al, 2005). A review of the existing tools has established that the existing agent development platforms are not stable enough as operational environments. Agent technology is still maturing; there is not a single development environment that can have been used on its own to implement all the components of an MAS architecture. This challenge can be addressed through leveraging on the extensibility of agents – components implemented in different environments can be integrated to function as a system. However, the extensibility that is inherent in agent-based systems cannot be fully exploited until there is a critical mass of researchers implementing components that can interface with each other. This problem is further compounded by the general lack of awareness within the construction industry of the potential of agent-based systems. A significant amount of the literature that is available focuses on technical issues such as agent architecture, MAS learning and negotiation protocols, yet gives limited information on lessons learnt in agent-based applications. Early adopters of agent technology should be encouraged to provide benchmarking and case studies of their experiences with implementing applications.

The implementation challenges cannot be totally attributed to agent paradigm being relatively new. Some early implementations of agent-based systems failed to take off because of how they were packaged. The value of agents can be best described in terms of business processes: software agents are just a subset of the new evolving technologies that promote efficiencies in the supply chain. Software agents on their own would not make a good business case for a commercial application. The approach adopted in implementing agent-based systems requires collaboration with researchers developing different types of computing applications used in construction. Agents should be presented as tools for enhancing the functionalities of VDC applications through automating features such as the ones exemplified in the proof of concept. It is therefore critical, even within an educational setting, to present agents as one part of the solution to the knowledge flow problems inherent in VDC activities.

It was previously established that a key argument for using the multi-agent approach is the agents' ability to cooperate in distributed problem-solving. A large-scale implementation of an agent-based system within a business process setting involves implementing a community of agents such as the one depicted in Figure 5.

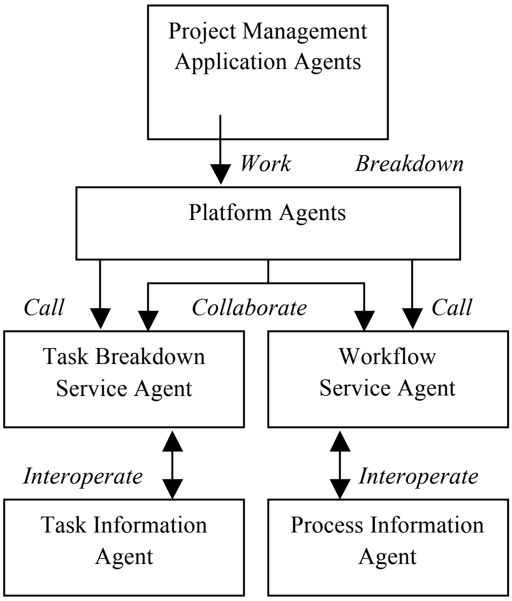

FIG 5: Cooperation in a Multi Agent Community
Adapted from Yaoqin et al. 2007

The agents in the system cooperate with each other, as shown in Figure 5. This example is based on the need for simulating workflow in a project. The relevant application agents propose a service request to platform agents, which in turn evokes the service agents that inter-operate with information agents to access, achieve, gather or transfer distributive information. The information is then communicated back to the platform agents. Although most of the discussion in preceding section focused on functions that would be handled by the information agent, interactions between ontologies and all the other agents can be modeled in a similar manner. Further research in this project will focus on deploying a multi-agent community exemplifying this type of cooperation among agents. Within the context of a multi-agent system, such information agents can also be used to support the seamless flow of information within a VDC application as discussed in the preceding section. This is an area which will be pursued further through additional research.

REFERENCES

AI³ - Adaptive Information, Adaptive Innovation (2010), Adaptive Infrastructure, The Sweet Compendium of Ontology Building Tools, http://www.mkbergman.com/862/the-sweet-compendium-of-ontology-building-tools/ [Accessed on 15th February 2010].

Assali, A.A.; Lenne, D.; Debray, B. (2007), KoMIS: An Ontology-Based Knowledge Management System for Industrial Safety, *Database and Expert Systems Applications, 2007. DEXA '07. 18th International Conference*, vol., no.,475-479, 3-7 Sept. 2007

Berners-Lee, T. (1989), Information Management: A Proposal, CERN, Geneva, Switzerland.

Berners-Lee, T, Hendler, J & Lassila, O. (2001) The Semantic Web, *Scientific American*, May 2001 Issue, [Online] Available from <http://www.sciam.com> [Accessed on 15th January 2002].

Bielawski, L. and Lewand, R., *Intelligent systems design: Integrating expert systems, hypermedia, and database technologies*, John Wiley & Sons Inc, New York (1991).

Brustoloni, J. C. (1991): Autonomous Agents: Characterization and Requirements, Carnegie Mellon Technical Report CMU-CS-91-204, Pittsburgh: Carnegie Mellon University.

McGuinness, D.L. (2003), Ontologies Come of Age, In Fensel, D., Hendler, J., Lieberman, H. and Wahlster, W. (eds). *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press.

Ding L., Liew, P.S., Maher, M.L., Gero, J. S. and Dogmuller, R. (2003), Integrating CAD and 3D Virtual Worlds Using Agents and EDM, In Chiu, M-L., Tsou, J-Y., Kvan, T., Morozumi M. and Jeng T-S (eds), Digital *Design: Research and Practice*, Kluwer, 301-323.

El-Diraby, T. E., Lima, C., and Fies, B. (2005), A Domain Taxonomy for Construction Concepts: Toward a Formal Ontology for Construction Knowledge, *Journal of Computing in Civil Engineering*, ASCE, Issue 19, No. 3, 394-406.

Ferber, J. (1999): *Multi-Agent System: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, An imprint of Pearson Education, UK, 9-11 & 53-56.

Fingar, P. (1998), A CEO's guide to e-commerce using Object-Oriented Intelligent Agent Technology, CommerceNet Research Report, No.98-19, USA.

FIPA Architecture Board (2001), FIPA Agent Software Integration Specification, 2001 [Online] Available from <http://www.fipa.org> [Accessed on 30 September 2002].

Gomez-Perez A., and Corcho, O. (2002), Ontology Language for the Semantic Web, *IEEE Intelligent Systems,* Vol 17, 54-60.

Gruber, T.R. (1995). Toward Principles for the Design of Ontologies used for Knowledge Sharing, *International Journal of Human-Computer Studies*, Vol. 43, Issues 4-5, November 1995, pp 907-928.

Guisheng, Y and Qiuyan, S. (2008), Research on Ontology-Based Measuring Semantic Similarity, In *Proceedings of the 2008 International Conference on Internet Computing in Science and Engineering*, pp250-253

Hendler, J., Berners-Lee, T. and Miller, E. (2002), Integrating Applications on the Semantic Web, *Journal of the Institute of Electrical Engineers of Japan*, Vol. 122 No. 10, pp 676-680.

Huhns, M.N. (2000) "Agent Teams: Building and Implementing Software," *IEEE Internet Computing*, Vol. 4, No. 1, pp. 91-93.

Issa M; Rankin J; Christian J; Pemberton E (2008), Lessons learned from the use of interactive workspaces for student team design project meetings, *ITcon Vol. 13*, *Special Issue Virtual and Augmented Reality in Design and Construction*, pp 674-690, http://www.itcon.org/2008/37

Issa, R.R.A, and Mutis, I. (2006), Ontology based Framework using the semantic web for addressing semantic reconciliation in Construction, In Smith I.F.C. (ed), *Intelligent Computing in Engineering Architecture*, Springer, Berlin, pp 348-367.

Jennings, N. R. (1999), Agent-based Computing: Promise and Perils, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden. (Computers and Thought award invited paper), pp 1429-1436.

Jennings, N.R. and Wooldridge, M. (1998), Applications of Intelligent Agents, In Jennings, N.R. and Wooldridge, M. (eds), *Agent Technology: Foundations, Applications, and Markets, Berlin: Springer Verlag*, pp3-28.

Kashyap, N. (1997), An Expert Systems Application in Electromagnetic Compatibility, Master's Thesis, University of Missouri-Rolla, USA.

Kim, K.-Y. Manley, D.G. Yang, H. and Nnaji, B.O. (2005), Ontology-based virtual assembly model for collaborative virtual prototyping and simulation, *Collaborative Technologies and Systems, 2005*, Proceedings of the 2005 International Symposium, pp 251 - 258 .

Kunz, J., and Fischer, M. (2009). Virtual Design and Construction: Themes, Case Studies and Implementation Suggestions, CIFE Working Paper #097, Stanford University, Stanford.

Lieberman, H. (1997), Autonomous Interface Agents, In Pemberton, S., editor, *Proceedings of the ACM Conference on Computers and Human Interface*, ACM/Addison-Wesley, pp 67-74.

Li, H., Li, W., Cai, Q and Liu, H. (2009), A framework of ontology-based Knowledge Management system, Computer Science and Information Technology, *ICCSIT 2009*, 2nd IEEE International Conference, pp 374 – 377.

Liao, K., Liu. Q., and Zhao. X. (2009), Ontology-Based Model of Mobile Knowledge Service for the Inspection of Construction Project, *Management and Service Science, 2009. MASS '09*, pp 1-5.

Luck, M., McBurney, P, Shehory, O. and Willmott, S, eds (2005), *Agent Technology Roadmap, A Roadmap for Agent Based Computing*, AgentLink III.

Maes, P. (1994), Agents that Reduce Work and Information Overload, In Bradshaw, J.M., (ed), *Software Agents*, AAAI Press/MIT Press.

Muramoto K, Jemtrud M, Kumar S, Balakrishnan B, Wiley D (2008), Emerging technologies in a tele-collaborative design studio between the Pennsylvania State University and Carleton University, *ITcon* Vol. 13, Special Issue Virtual and Augmented Reality in Design and Construction , 660-673, http://www.itcon.org/2008/41.

North , M.J. and Charles M. Macal, C.M. (2007), *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation*, Oxford University Press, Inc., New York, NY, 2007

Nwana, H.S. (1996), Software Agents: An Overview, *The Knowledge Engineering Review.* Vol. 11, No.3, 205-244.

O'Brien, P. D., and Nicol, R. (1998), C. FIPA - towards a standard for software agents. *BT*

Technology Journal, Vol.16, no.3, July, 1998, pp 51.

O'Malley, S.A. and DeLoach, S. 2002, Determining When to Use an Agent-Oriented Software Engineering Paradigm, Agent-Oriented Software Engineering II, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp 188-205.

Obonyo, E.A., Anumba, C.J. and Thorpe, A. (2004), Specification and Procurement of Construction Products, A Case for an Agent-based System. *International Journal of IT in Architecture, Engineering and Construction*. Vol.2, No. 3, pp 204-215.

Obonyo, E.A., Anumba, C.J. and Thorpe, A. (2005), APRON: an agent-based specification and procurement system for construction products. *Engineering, Construction and Architectural Management*. Vol. 12, No. 4, 329-350.

Pan, J. (2006), Construction Project Information Management in a Semantic Web Environment, PhD Thesis, Loughborough University, Loughborough, October.

Paranuk, H. V. D. (1996), Applications of Distributed Artificial Intelligence In Industry, O'Hare, G.M.P. and Jennings, N.R. (eds), *Foundations of Distributed Artificial Intelligence*, Wiley Inter-Science.

Poslad, S., Buckle , P., Hadingham, R. (2000), Open Source, Standards and Scaleable Agencies, Revised Papers from the International Workshop on Infrastructure for Multi-Agent Systems: Infrastructure for Agents, *Multi-Agent Systems, and Scalable Multi-Agent Systems*, pp 296-303.

Radeke, E., ed (1999), *Final Report, GENIAL Global Engineering Networking Intelligent Access Libraries*.

Sanchez, D.M., Cavero, J.M. and Martinez, E.M. (2007), The Road toward Ontologies, In Sharman, R, Kishore, R and Ramesh, R. (eds), *Ontologies: a handbook of principles, concepts and applications in information systems*, New York : Springer, pp 3 - 20

Sankhwal, S., Rasaratnam,R. and Yiu, L. (2000) Article 1: Multi-Agent Systems, In SURPRISE 2000 Surveys, Department of Computer Science, Imperial College UK.

Stephens, J., Dimtas, N., Lima., C., Steinmann, R., Woestenenk, K., Bohms, M. and Tolman, F. (2002): *ECommerce and eBusiness in the Building and Construction Industry, Preparing for the Next Generation Internet, Final Report*.

Sycara, K.P., (1998), Multiagent Systems, American Association for artificial Intelligence.

Thiels, L., Bragoni, R. and Shamsi, T.A. (2002), Efficient Bidding and Procurement in the Tile Industry – Practical Trading Tools and Broker Services for the Exchange of Product Characteristics, e-bip Final Report.

Ugwu, O. O., Kamara, J. M., C. J. Anumba and Leonard, D. (2002), Electronic Procurement of Construction Products, *International Journal of Service and Technology and Management*, Vol.3, No.2, pp 222-237.

Wooldridge, M., 2002, *An Introduction to Multiagent Systems*. John Wiley and Sons Ltd, February 2002.

Yu, S. and LI Zhiping, L., 2009, Ontology-based Domain Knowledge Representation, Proceedings of 2009 4th International Conference on Computer Science & Education, pp 174 – 177.