A COMPUTING CURRICULUM WIKI: ANALYSIS AND MODELLING USING THE MAS-CommonKADS AGENT-ORIENTED METHODOLOGY

ADELINA TANG
Sunway University
No. 5, Jalan Universiti, Bandar Sunway
46150 Petaling Jaya, Selangor Darul Ehsan
Tel: +60(3)74918622 Ext.3314
Fax: +60(3)56358633
adelinat@sunway.edu.my

ANGELA LEE
Sunway University

ABSTRACT

Curriculum development, maintenance and management are time-consuming and labour-intensive activities resulting from countless feedback-rework cycles. The frequency of such activities tends to increase owing to the accelerated nature of advances in Computing. It is proposed that an existing Computing Ontology be adapted to facilitate these activities by developing a common vocabulary for all Computing disciplines to realize an online Curriculum Wiki facility. The operations of the Wiki would be implemented through ontological agents. This article presents insights into the modelling process of various user-initiated Wiki tasks using the MAS-CommonKADS Agent-Oriented Methodology.

Keywords: computing ontology, curriculum wiki, agent, MAS-CommonKADS

INTRODUCTION

Curriculum development, maintenance and management are time-consuming and labour-intensive. Meetings, discussions, differences of opinion and countless feedback-rework cycles serve to make these activities acutely problematic but totally necessary in any academic institution. Furthermore, curriculum updating activities are expected to be carried out fairly regularly within a three-year timeframe owing to the accelerated nature of advances in Computing. It is proposed that an existing Computing Ontology be adapted to facilitate these activities by developing a common vocabulary for all Computing disciplines as a first step.

To be specific, Cassel's Computing Ontology project (Cassel, Sloan, Davies, Topi, & McGettrick, 2007) is utilized as the Ontology on which the Curriculum Wiki is based. The Wiki project introduces further enhancements so as to adapt Cassel et al.'s work to enable the sharing of stakeholders' feedback and to aid curriculum developers. To this end, the project proposes that ontological agents be utilized to facilitate the Wiki operations. These agents are modelled using the MAS-CommonKADS Agent-Oriented Methodology (Medina, Sánchez, & Castellanos, 2004) which ensures that all possible use cases and scenarios are fully analyzed from various perspectives. This paper provides an insight into the modelling process and highlights the complexities therein.

LITERATURE REVIEW

Computing Ontology

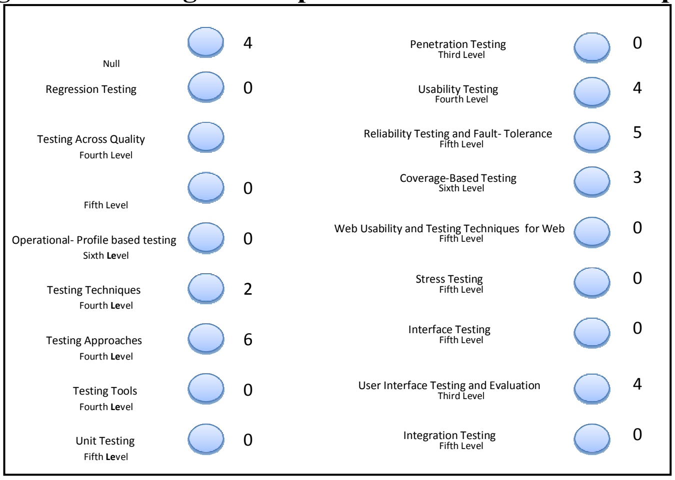
An Ontology facilitates the sharing of knowledge. It is a specification of a representational vocabulary for a shared domain of discourse, as defined in Gruber's seminal work (Gruber, 1993). The shared domain of discourse would consist of classes, relations, functions, and similar objects of interest. Ontologies provide the basic structure around which knowledge bases can be built (Swartout & Tate, 1999). Ontological engineering activities include philosophy, knowledge representation formalisms, development methodologies, knowledge sharing and reuse, information retrieval from the Internet or any online repositories, to name a few. It provides a systematic design rationale of a knowledge base according to the context of interest (Devedžić, 2002). Berners-Lee and colleagues in their famous paper (Berners-Lee, Hendler, & Lassila, 2001) included Ontologies as the important third basic component of the Semantic Web, stating that Web Ontologies typically consist of a taxonomy that defines classes of objects and relations among them, and a set of inference rules. Inference rules feature prominently in the manipulation of terms that provide meaning to the human user. Furthermore, the combination of the taxonomy and the inference rules provide equivalence of meanings from two or more disparate information sources.

Cassel et al (2007) proposed their Computing Ontology project with sponsorship from the United States National Science Foundation (US NSF), the Association for Computing Machinery (ACM), and the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS). The authors compressed the five distinct fields, Computer Engineering, Computer Science, Information Systems, Information Technology, and Software Engineering, into one generic computing field. This was to facilitate the development of a single Computing Ontology. In their project, the primary objective was to connect the comprehensive list of typical computing topics with curriculum development and course-planning activities. Thereafter, a prototype system for matching course topics and outcomes would emerge. Cassel, Davies, LeBlanc, Snyder and Topi (2008a) proposed a web-based utility to enable a course developer to select or create outcomes as well as to select suitable topics that could achieve those very outcomes.

Upon further scrutiny, it is proposed that some enhancements be introduced into Cassel's current project, so as to adapt it to fulfil this project's objectives without having to create it from scratch. The screenshots in Figures 1 and 2 illustrate the limitations of the visualizations of the existing Computing Ontology (Cassel, Davies, LeBlanc, Snyder, & Topi, 2008b). The authors used "Flash" screens to illustrate a drill-down from top level concepts to subsequently more specific concepts.

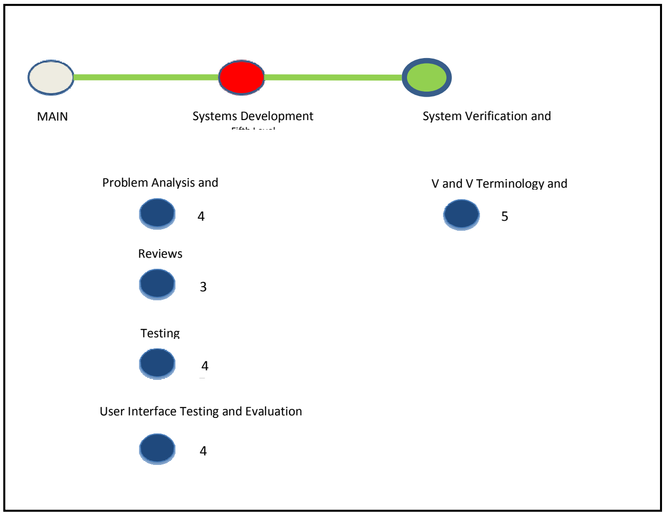
Note that these illustrations do not indicate whether such concepts would be taught in the same or in separate courses. The "Flash" screens also do not interface with any other information source. This limits their utility in supporting actual curriculum maintenance and management activities. For instance, no outcomes are associated with any of the concepts. While the depth of coverage is indicated, the breadth of coverage across different courses and possibly even different Computing disciplines, is not indicated clearly. The authors also admit that there might be too many top level concepts and that these should be reduced.

Figure 1. "Testing" Concept Reveals All Related Concepts.



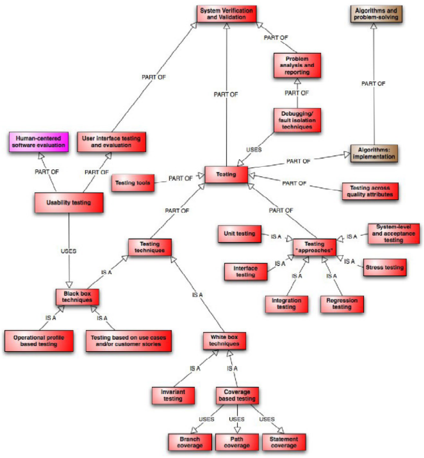
A further functionality is revealed by searching for the "Testing" concept, the result of which is displayed in Figure 1. Although there appear to be numerous "Testing"-related concepts, Figure 2 shows that not all reside within the major concept of "System Verification and Validation". Additionally, Figure 1 does not indicate which other major concepts might contain the other "Testing" concepts. This alludes to the issue of solely adopting the *horizontal* approach in which concepts exist across several courses, without considering the *vertical* approach which views concepts as resident within a specific course.

Figure 2. "Testing" Concepts Reside within the "System Verification and Validation" Major Concept that Belongs to the "Systems Development" Top Level Concept.



Cassel et al. (2008a) pointed out that their Computing Ontology "provided a tool for faculty and students to use in determining how much of a given topic is appropriate in a particular context". They added that the role of the Ontology was to show the relationships of concepts to facilitate decision-making, and provided a hand-drawn graphical representation of the "Testing" concept (see Figure 3) to strengthen their arguments.

Figure 3. Hand-drawn Graphical Representation of the "Testing" Concept.



Owing to the limitations of the existing Computing Ontology, this paper proposes to create a Computing Curricula Repository (CCR) to provide similar graphical representations as an added utility in aiding curriculum developers to make informed decisions. It is envisaged that the adapted Computing Ontology would play the role of a knowledge base that provides the necessary information to realize visualizations of concepts that exist across disparate courses as well as within a specific course.

Ontologies and Agents

If one views the different computing disciplines as different databases, i.e. the Computer Science database, Information Technology database, etc., grouping them together would then make them a set of federated databases. An alternative would be to organize the curriculum in terms of their respective years of study, i.e. Year 1 Common Core, etc. Regardless of the preference, obtaining information from different databases represent different challenges.

In Medina et al. (2004), the authors proposed utilizing agents and Ontologies to retrieve information from a set of federated digital libraries. They adopted the MAS-CommonKADS Agent-Oriented Methodology (AOM) to model their agents. MAS-CommonKADS is extended from the CommonKADS methodology that uses some OO concepts and techniques. However, it ultimately reflects its Knowledge Acquisition, Engineering and Management roots from the ESPRIT IT Program (CommonKADS, 1995).

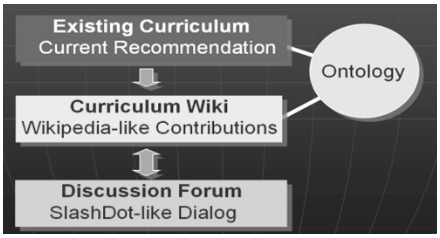
The Foundation for Intelligent Physical Agents (FIPA) provides an ontology-based description of the knowledge model, but leaves the understanding, manipulation and the internal agent memory model to the developer's discretion and preference (FIPA, 2001). Ultimately, the application of XML, a key component of the Semantic Web (Berners-Lee et al., 2001), provides intelligent access to heterogeneous and distributed sources. In essence, (Fensel, 2001) pointed out that agents operate as mediators between user needs and information resources.

This paper proposes to adopt the MAS-CommonKADS AOM to model the agents and their accompanying processes so as to facilitate the user-initiated Wiki tasks owing to the success reported by the authors in modelling agents and Ontologies (Medina et al. 2004).

METHODOLOGY

The scheme proposed by Cassel et al. (2008a), shown in Figure 4, is adopted for the CCR. The "Existing Curriculum" is fixed and not editable by the larger Computing community. The "Curriculum Wiki" is an editable environment which allows suggested additions and revisions to be introduced into a copy of the existing curriculum, i.e. a working/discussion copy. These activities are made transparent to all members of the Computing community. The third component, the "Discussion Forum", enables discussions regarding the aforementioned additions and revisions. Note that the middle section, the Curriculum Wiki, is the focus of this paper.

Figure 4. Computing Ontology Development Scheme.



A study by Cassel et al. (2008c) was initiated to determine the ontological agent characteristics and tasks to be undertaken by them as part of the overall Curriculum Wiki project requirements. Thereafter, the MAS-CommonKADS AOM was used to model the ontological agents that are tasked to extract relevant resources. The AOM consists of three phases: Conceptualization, Analysis, and Design.

In the Conceptualization phase, a preliminary description is elicited through the application of used cases/scenarios which are formalized with Message Sequence Charts (Iglesias, Garijo, González, & Velasco, 1998). Briefly, its Analysis phase consists of the following models:

- *Agent modelling* - Identification and description of initial instances of agents. A textual template is used for each agent that includes name, type, role, position, description, services, and so on.

- *Task modelling* - A top-down approach is used to decompose tasks. In the Curriculum Wiki, the tasks performed by the agents are already decomposed in the Agent Model. Since the Task Model is similar to the Agent Model, its discussion has been omitted.
- *Expertise modelling* - It consists of the development of the application knowledge and problem-solving knowledge.
- *Coordination modelling* - It defines communication channels, construction of a prototype and coordination protocols.
- *Knowledge modelling* - It models the reasoning of the agent in a domain and the inferences of the environment. In the Curriculum Wiki, the reasoning and environmental influences are already captured in the Expertise Model. Since the Knowledge Model is similar to the Expertise Model, its discussion has been omitted.
- *Organization modelling* - It models the static or structural relationships between agents. This is not applicable in the Curriculum Wiki context as the Coordination Model more accurately captures the relationships among agents by modelling their communication and coordination characteristics.

The next section will detail the Conceptualization and the Analysis phases which have been completed. The Design phase and the development work are targeted for completion within the next four months.

RESULTS

Table 1 summarizes the relevant agents and their Performance, Environment, Actuators and Sensors (PEAS) characteristics.

Table 1. Curriculum Wiki Agents

Agent Type	Performance	Environment	Actuators	Sensors
Manager	Act as a communicator among all the agents. It is also able to accept user request and forward each user request to the predictor/historian.	Virtual	Perform interaction among agent	Keyboard, Mouse
Retriever	Minimizes the delivery time and avoids human intervention. For example, one-time data entry would help reduce	Virtual	Display results of search and learning outcomes	Keyboard , Mouse

	the chances of re-introducing new errors. This agent is also able to “post and retrieve” information.			
User	Send request, post messages.	Virtual, Administrative and Academic Staff, Curriculum Steering Committee	Prompt instructions	Keyboard, Mouse
Predictor	Consists of a prediction tool which allows the system to keep track of user activity and stores every request in order to generate a user request pattern.	Virtual	Display result	Keyboard

Four user requests were identified:
1. UR#1: Search for Learning Outcomes from Course Code/Course Name
2. UR#2: Search for all related concepts from Keyword(s)
3. UR#3: Request for Course Exemption
4. UR#4: Request for Change in Course Syllabus

Each of these user requests was analyzed using the MAS-CommonKADS AOM. The relevant phases include:
1. Conceptualization Phase
   a. Conceptualization Modelling
2. Analysis Phase
   a. Agent Modelling
   b. Coordination Modelling
   c. Expertise Modelling

As user requests UR#1 and UR#2 were found to yield similar results, this paper will only detail UR#1. UR#3 and UR#4 will include only details that are different from those stated earlier. Tables 2 – 4 describe the results of the analysis of user requests, UR#1, UR#3 and UR#4, using the MAS-CommonKADS AOM.

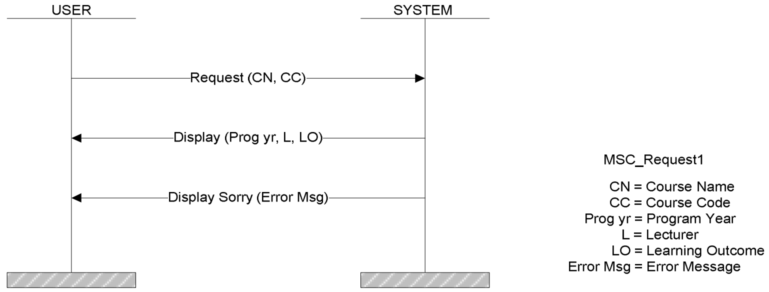
Table 2. UR#1: Search for Learning Outcomes from Course Code/Course Name.

<u>Phase 1 : Conceptualisation Modelling - Message Sequence Chart</u>

In this phase, use cases/scenarios were used to assist the understanding of all the possible requirements. This helped to identify all pertinent agent interactions.

Steps:

1. Key in Course Name or Course Code
2. Display Programme Year, Level and Learning Outcomes



The user would need to key in the course code or course name in order for the system to display the programme year, level and learning outcomes. If the user keys in a wrong or invalid course code, the system will generate an error message telling the user that the course code is an invalid code.

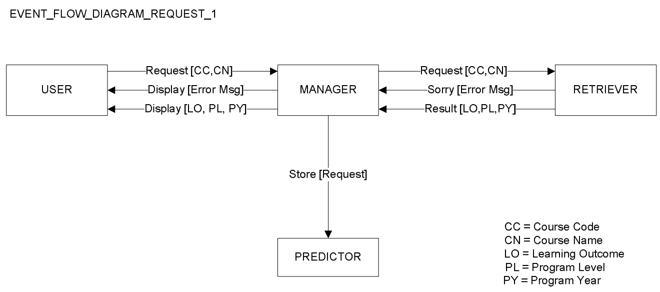
<u>Phase 2: Agent Modelling</u>

The agents identified include the Manager, the Retriever and the Predictor. In this case, the User is the initiator of the request.

Name of Agent	Role	Type	Services
Manager	Interact with the user by accepting the request of user, and displaying the results.	User Interface	Process user request and collect the results to be displayed to the users.
Retriever	Retrieve data from the server. In this case, the agent will retrieve the learning outcomes and programme year and level.	User interface and Computing Ontology	Conduct a search in order to retrieve the data from server.
Predictor	Store the results and records of all user requests.	Computing Ontology	Store all user tasks, useful when the task is repeated.

Phase 3: Coordination Modelling - Event Flow Diagram

The event flow diagram shows a typical interaction among the four agents. This diagram explains and demonstrates the flow of communication between each agent and its outcomes. The interchange of data is shown in squared brackets.



Phase 4 : Expertise Modelling

This model describes the knowledge needed by each agent to achieve its goals.

Agent	Goals	Knowledge Needed
Manager	- Display results of user request - Communicate with Retriever	- User request able to forward requests to the following agent
Retriever	- Conduct search - Retrieve information from the server - Communicate with server	- Communication from Manager event - Data request from Manager event
Predictor	- Store user task - Store a pattern of user requests	- Every interaction that occurs between Manager and Retriever - User request

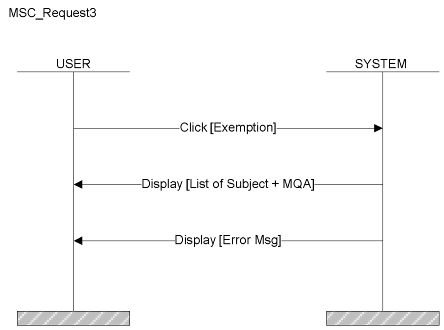
Table 3. UR#3: Request for Course Exemption.

Phase 1 : Conceptualisation Modelling - Message Sequence Chart

Steps:

1. Click on "Exemption" icon
2. System perform check
3. Display the list of all possible course

All exemptions are only applicable for Year 1. The MQA document will be made available to the user once the system is able to find the relevant course to be exempted.

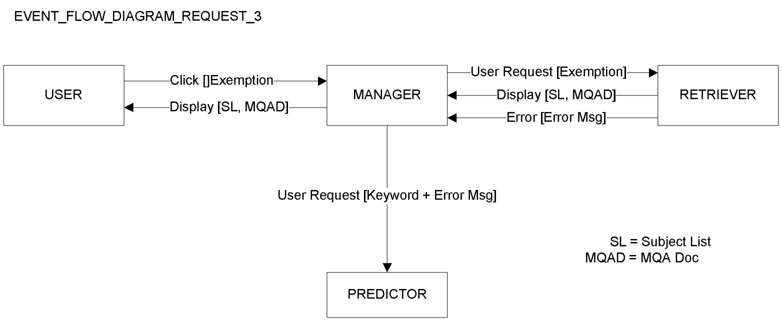


Phase 2: Agent Modelling

This model is similar to that of UR#1 with the exception of the Retriever and Predictor details.

Name of Agent	Role	Type	Services
Retriever	Search possible keyword on MQA document	Computing Ontology	- Perform matching. - Extract relevant MQA document from the server. - Alert Predictor about new request.
Predictor	Store request history	Computing Ontology	Facilitates repeating request

Phase 3: Coordination Modelling - Event Flow Diagram



Phase 4 : Expertise Modelling

Agent	Goals	Knowledge Needed
Manager	- Display results	- User request (Keyword)
Retriever	- Conduct search - Retrieve information from the server	- Communication from Manager event - Data request from Manager event
Predictor	Store user task - Store a pattern of user requests	- User request

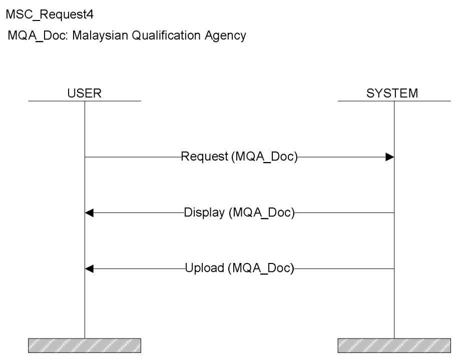
Table 4. UR#4: Request for Change in Course Syllabus.

Phase 1 : Conceptualisation Modelling - Message Sequence Chart

Steps:

1. User download MQA document
2. Edit on their perusal
3. Upload document of text reference and store in Curriculum Wiki.

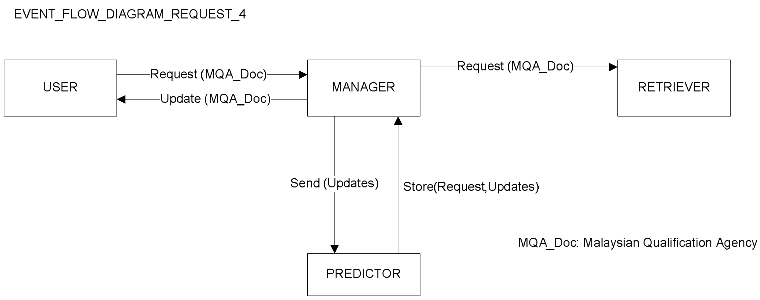
If the user wants to make changes to the syllabus, he can download the MQA document from the server. After that, the user can then upload the document to the Curriculum Wiki. The Wiki acts as the platform where the user and other stakeholders interact. Typically, a discussion forum is automatically initiated in the Curriculum Wiki to accommodate such virtual communications. The discussion timeline for all stakeholders' comments is approximately 2 weeks. Stakeholders comprise course specialists and Steering Curriculum Committee members. The Steering Curriculum Committee is tasked with approving/rejecting such changes. Upon approval, the system is able to replace the existing MQA document with the working copy. All activities and accompanying information are logged by the Predictor.



Phase 2: Agent Modelling

Name of Agent	Role	Type	Services
Manager	Interact with the user by accepting the request of user and displaying the results.	User Interface and Curriculum Wiki	Process user request and collect the results to be displayed to the users.
Retriever	Retrieve data from the server. Document of MQA will be sent back to the user to make changes.	User Interface and Ontology Computing	- Conduct a search in order to retrieve the data from the server. - Perform search on the relevant MQA document
Predictor	Store the results and records of all user requests.	Computing Ontology and Curriculum Wiki	Record all requests and comments on the system.

Phase 3: Coordination Modelling - Event Flow Diagram



The event flow diagram shows the user requesting a MQA document from the Manager. The Manager retrieves the documents from the Retriever. Manager and Predictor record all user requests for future use. The Curriculum Wiki acts as the platform where the user and other stakeholders interact. This reduces the number of meetings required in curriculum planning. Every user request is recorded by the Predictor. A discussion forum is initiated in the Curriculum Wiki to accommodate such virtual communications. The Steering Curriculum Committee is tasked with approving/rejecting changes through the forum. Upon approval, the system is able to replace the existing MQA document with the working copy. An acknowledgement is sent to alert the user that an approved change in the course syllabus has been updated.

<u>Phase 4 : Expertise Modelling</u>

This model is similar to that in UR#3 with the exception of the Predictor which requires knowledge of the Discussion Forum in addition to the existing knowledge of the user request.

DISCUSSION

Benefits of the MAS-CommonKADS AOM Models

The Conceptualisation phase gave a rough idea of the agent interactions that could take place (Iglesias et al., 1998). These agent interactions were refined during Coordination Modelling in the Analysis phase. The resulting refinements specified the data/knowledge interchanged and the speech-act of each interaction seen in the Event Flow Diagrams, Coordination Modelling in Tables 2 – 4. Hence, the refinements provided useful details to the developer to facilitate prototyping.

In addition, Agent Modelling ensured accurate description of the agent roles and the services that each is responsible to provide. It also defined the information source from which the agent should retrieve. This model provides the full set of characteristics of individual agents.

Finally, Expertise Modelling determines the reasoning capabilities of the agents in carrying out their specified tasks and in achieving their goals (Iglesias et al., 1998) (see Expertise Modelling in Tables 2 – 4). One benefit of using this model is that previously developed instances of the expertise model could be re-used and adapted to new characteristics of the agents should requirements change.

Future Work

As a result of the Conceptualisation and Analysis phases, the initial agents and their corresponding characteristics have been determined. The next stage of this project would realize the Design phase of the MAS-CommonKADS AOM which includes the agent network design, the agent design and the platform design. Thereafter, the development of the ontology and the Computing Curricula Repository would begin.

CONCLUSION

In conclusion, it has been shown that software ontological agents can be modelled successfully by the MAS-CommonKADS Agent Oriented Methodology. These agents are created to facilitate online curriculum development, maintenance and management tasks of the Computing Curriculum Wiki.

REFERENCES

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The semantic web. *Scientific American*, *284*(5), 34-43. Retrieved from http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2. [Site visited 15th July 2009].

Cassel, L. N., Sloan, R. H., Davies, G., Topi, H., & McGettrick, A. (2007). The Computing Ontology project: The computing education application. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (pp. 519-520). Covington, Kentucky, USA: ACM.

Cassel, L. N., Davies, G., LeBlanc, R., Snyder, L., & Topi, H. (2008a). Using a computing ontology as a foundation for curriculum development. In *Proceedings of the Sixth International Workshop on Ontologies & Semantic Web for E-Learning* (pp. 21-29). Montreal, Canada.

Cassel, L. N., Davies, G., LeBlanc, R., Snyder, L., & Topi, H. (2008b, August). The Computing Ontology Project. Retrieved from http://what.csc.villanova.edu/~harsha/flash/August08/augustVersion6.swf. [Site visited 8th July 2009].

Cassel, L. N., Davies, G., LeBlanc, R., Snyder, L., & Topi, H. (2008c, August). The Computing Ontology Project. Retrieved from http://what.csc.villanova.edu/twiki/bin/view/Main/OntologyProject. [Site visited 8th July 2009].

CommonKADS (1995). What is CommonKADS? Retrieved from http://www.commonkads.uva.nl/frameset-commonkads.html [Site visited 7th August 2009].

Devedžić, V. (2002). Understanding ontological engineering. *Communications of the ACM*, *45*(4ve), 136-144.

Fensel, D. (2001). *Ontologies: A silver bullet for knowledge management and electronic commerce*. Berlin: Springer.

Foundation for Intelligent Physical Agents (FIPA) (2001, August). Ontology Service Specification. Retrieved from http://www.fipa.org/specs/fipa00086/XC00086D.html [Site visited 7th August 2009].

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199-220.

Iglesias, C. A., Garijo, M., González , J. C., & Velasco, J. R. (1998). Analysis and design of multiagent systems using MAS-CommonKADS. In M. P. Singh, A. S. Rao, & M. Wooldridge (Eds.), *Lecture Notes in Computer Science; Vol. 1365. Intelligent Agents IV, Agent Theories, Architectures and Languages: Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages* (ATAL-97), Providence, Rhode Island, USA, July 24-26, 1997 (pp. 313-327). London: Springer.

Medina, M. A., Sánchez, A., & Castellanos, N. (2004). Ontological agents model based on MAS-CommonKADS methodology. In *Proceedings of the 14th International Conference on Electronics, Communications and Computers (CONIELECOMP'04)* (pp. 260-263). Veracruz, Mexico. Washington, DC: IEEE Computer Society.

Swartout, W., & Tate, A. (1999). Ontologies, guest editors' introduction. *IEEE Intelligent Systems*, *14*(1), 18-19.