12-31-2007

Agent-oriented e-Learning Process Modeling

Hokyin Lai
City University of Hong Kong

Huaiqing Wang
City University of Hong Kong

Minhong Wang
The University of Hong Kong

Agent-oriented e-Learning Process Modeling

Hokyin Lai
Department of Information Systems
City University of Hong Kong
isjlai@student.cityu.edu.hk

Huaiqing Wang
Department of Information Systems
City University of Hong Kong
iswang@cityu.edu.hk

Minhong Wang
Division of Information and Technology Studies
The University of Hong Kong
magwang@hkucc.hku.hk

Abstract

Students learn through a series of systemic and innovative learning activities instead of only learning resources. However, most e-Learning systems nowadays are resources-oriented instead of process-oriented. In addition, existing systems are designed without adequate pedagogical principles and provide little capabilities to cater for different student's individual needs. This paper investigates the possibility of applying process-oriented approach to design, enact, control, and refine the e-Learning processes. Furthermore, agent technology is applied to deal with dynamic situations. A process-oriented and agent-based e-Learning environment is developed, with case scenario to demonstrate the usability of the approach.

Keywords

e-Learning, Process Modeling, Process-oriented, Pedagogical Rule, Agent Oriented Modeling, Intelligent Agent, Ontology

Introduction

Constructivism, one of the famous learning theories, addressed that learning should be process-oriented instead task-oriented. Students learn through a series of systemic and innovative learning activities instead of purely learning resources. However, most of the available e-Learning tools, like BLACKBOARD and WebCT, are resource-oriented rather than process-oriented. They act as repositories for learning objects only. Without clear instruction, students would not know the steps to go through the learning process. Instead, they are just informed how to download the learning objects from there. This problem reveals the need of a Process-oriented e-Learning Model to systematically organize and manage learning tasks under pedagogical principles.

Referring to the e-Learning Process Design approach, instructor first identifies the learning goals, then designs learning activities by selecting suitable learning tasks, combining them in an appropriate sequence, mapping corresponding learning resources to each task, and select a right system tool (or system functions) to support both individual tasks and the whole learning process. When the design is set, the tasks can then be put into execution state, such that students can access to the right resources at the right time.

Process Management has been widely used in the business world. It makes use of methods, techniques, and software to design, enact, control, and review operational process that involving humans, organization, applications, documents, and other related sources of information (Weske, 2004). Both business process and learning process are processes in nature. The Business Process Management (BPM) lifecycle is quite similar to that of the e-Learning Process. And this paper is aimed to investigate the possibility of applying BPM approach to manage e-Learning process.

Considering the capability to cater for individual student's needs in dynamic learning situations, software agents are applied to provide adaptive learning process. Since each individual student is unique in nature, they have different learning styles because of their different backgrounds and/or preferences. As a result, they use the e-Learning tool in different usage patterns, which may affect the effectiveness of the learning process. As pointed out by Hawryszkiewycz (2005), software agents can actively assist business process participants to select the most appropriate engagements to fill in this kind of knowledge gap.

This paper aims at tackling the above-mentioned problems by building a process-oriented and agent-based e-Learning model. First, literature reviews on the concepts of learning theory, e-learning process model, intelligent agents for e-Learning and Business Process Management are covered. Second, after identifying the problems and suitable technologies, an Agent-oriented e-Learning Process Model is proposed. Explanations on the agent properties, relevant knowledge facilities and the agent architecture are also discussed. Third, a case scenario is demonstrated to illustrate the usability of this approach. Finally, an evaluation of the features between the Agent-based Process-oriented e-Learning Model and the traditional model is discussed.

Literature Review

Learning Theory

Constructivism, one of the famous learning theories that addressed learning is an active process in which meaning is developed based on knowledge and experiences. Experience is earned through the execution of the set of learning processes with social negotiations with other participants.

The problem-solving heuristic process defined by Krulik and Rudnick (1996) is composed of five individual tasks which are in the order of: Information gathering; Organization of Information; Evaluation of alternatives; Execution of the most appropriate alternative; and Reflection on the outcome. The tasks work closely with each other in sequence. Additional social interaction has been incorporated in the model to ensure the cohesiveness between the tasks as well as between the participants. Instructions during learning process are crucial to the process outcomes (Proctor, 1984). Students with instructions could achieve a more desirable learning outcome. This example revealed that learning process is composed of an ordered set of learning tasks together with the aid of social interactions in order to accomplish a predefined pedagogical goal.

E-Learning Process Model

Even though a lot of e-Learning tools are now available in the market, most of them like BLACKBOARD or WebCT are designed for course content management instead of learning process management (Marjanovic, 2005). Instructor uploads the learning resources and then students download them without any guidance. The system designers only treat the learning resources as some discrete and independent items. An exception is the latest educational technology standard SCORM (Sharable Content Object Reference Model) (ADL, 2004), which supports the alignment of the learning resources with sequencing and the reuse of learning objects. However, the focus of SCORM is to provide a standard that enable interoperability of reusable learning objects, rather than provide process support.

Another problem of the existing tools is that most of them do not provide adaptive feature. One should know that the differences in learning style, learning background and learning expectation in a same class are quite obvious. These factors affect the behaviors that students deal with the e-Learning tool. Therefore, a lot of educational researches concern about adaptive learning and Proctor (1984) strongly believes that the overall learning outcomes can be improved by adaptive learning. There is a prominent demand in this area.

Agent-oriented Modeling for e-Learning

Wooldridge and Jennings (1995) defined an agent as a computer system that work in an autonomous way and can work under dynamic environment in order to achieve the goals defined by the system designer. Furthermore, they also claimed that agents can work interactively with each other without human intervention. The interactivity of the agents opens a communication channel for the participants (i.e. the instructors and students) to deal with the e-Learning system. The presence of agents stimulates the human aspect of instruction in a natural and valid way than other computer-based methods. The representation of knowledge in the agent-based architecture can support students' learning activities adaptively (Xu and Wang, 2006).

In the past, agents were mostly designed as personal assistants, human-system mediators, and so forth (Aroyo and Kommers, 1999). The need for an explicit presentation of course content has been continuously growing, such that concepts like student model and pedagogical diagnosis have been widely discussed. Most of the agent-based e-Learning models possess reasoning power, such that they can provide dynamic adaptive features based on the student profiles and pedagogical instructions. However, such a design involves complicated knowledge exchanges among agents. This would affects the efficiency of the model, thus, multi-agent architecture can overcome this problem. However, there exists a challenge in designing a comprehensive e-Learning model. Continuous investigations on the effectiveness of e-Learning model in real world are needed as well.

Business Process Management

Business Process Management (BPM) enables an end-to-end visibility and control over all business processes in one or even more than one companies. The process model uses of methods, techniques, and software to design, enact, control, and review operational process that involving humans, organization, applications, documents and other related source of information (Weske, 2004). The increasing adoption rate of the process-oriented technologies like Workflow Management System (WfMS) can reflect the regard to the process modeling. Process modeling can enhance the interoperability and cohesiveness between business processes (Wang et al. 2005). However, the external environment in business world is ever changing. Realizing the need to provide sufficient flexibility and adaptability in business processes, many researchers are investigating adaptive workflow techniques (Wang and Wang, 2006).

As defined by (Weske et al. 2004), business process modeling is composed of four main phases.

(1) **Design phase:** Essential resources to the process, which include humans, organizations, applications, documents, and other sources of information, are assembled first. Then, mappings between the methods, techniques, and technical infrastructure and the resources are performed to output a conceptual model.

(2) **Configuration phase:** Service specification, interface definition, and workflow specification are done in order to transform the high-level conceptual model into an executable one.

(3) **Enactment phase:** The process is executed by deploying the services.

(4) **Diagnosis phase:** The production performance is evaluated. In addition, process can be further tuned to become more robust.

Agent-oriented e-Learning Process Model

In this section, an agent-oriented e-Learning Process Model is discussed. This model is designed basically based on the pedagogical requirements derived by Constructivism. The instructional goal of the model is to equip students with an increasing repertoire of strategies, approaches, and familiar course context. Throughout the whole learning process, involvement of instructor is inevitable. This model also offers adaptive features. The expected contribution is that the learning outcome can be enhanced.

E-Learning Process

Referring to the centered approach derived by Marjanovic (2005), a typical e-Learning Process is modeled (Figure 1). The model is formed using five basic steps: (1) to identify the pedagogical goals and pedagogical objectives of the course. (2) to identify some learning activities that can guide the student to achieve the goals. This step can be further divided into few sub-steps, which are: selecting individual learning tasks, grouping related tasks into one learning process with an order, mapping available learning resources to each individual task, and then selecting appropriate system functions to support them. In addition, in order to maintain the interoperability between tasks, some extra tasks can be added as well. Just like the social interaction used in the problem-solving heuristic process mentioned in previous section. (3) to execute the learning activities according to the predefined sequence. (4) to closely monitor the execution of the learning activities. (5) to review the existing model based on the learning progress of students. Amendments on the existing learning activities are crucial to support the adaptive learning feature. Since adaptive learning is a recursive process, so step 2 to 5 is repeated until the course is about to end or no further improvement can be done.

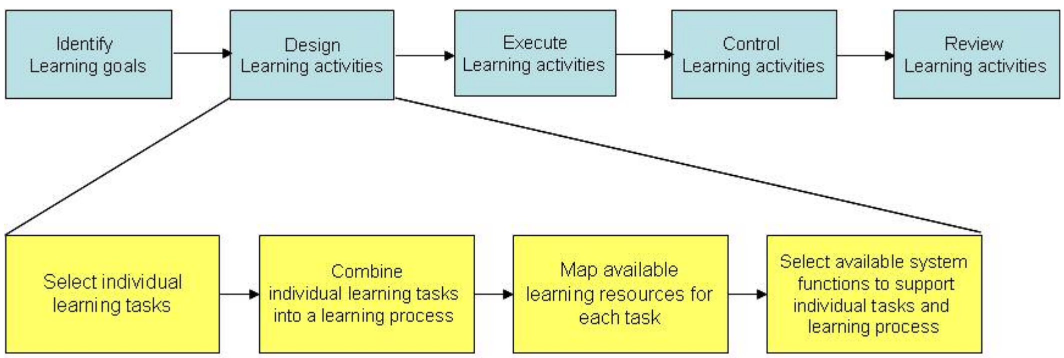

Figure 1. Typical e-Learning Process Formation

For example, an adaptive e-Learning process for a 4-week introductory course to Information Systems is shown in Figure 2. The process is the expected outcome of the second step (Design Learning activities) in Figure 1. The learning tasks of the process are interrelated. Each learning task links with corresponding learning resources and system functions. Once the process is set, student can start learning from the most preceding task.

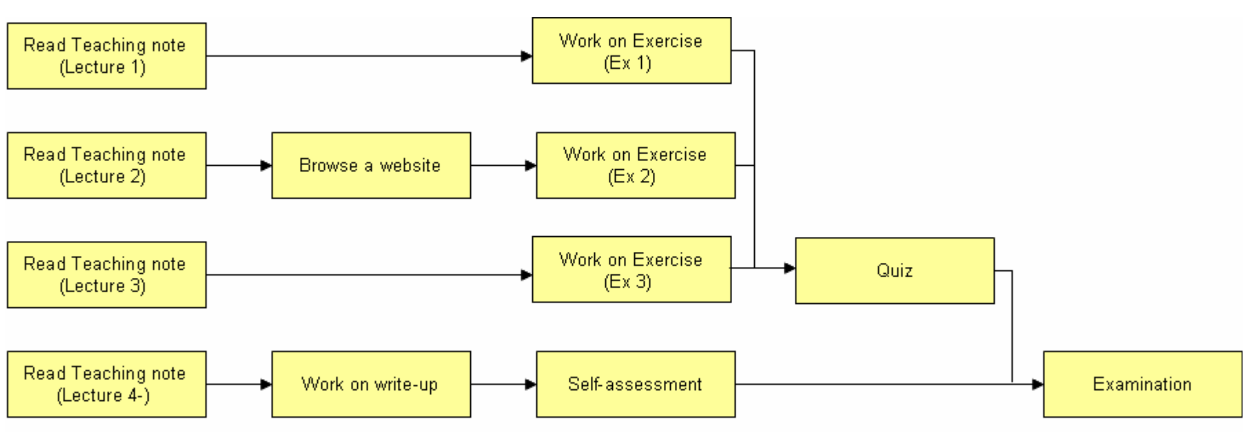

Figure 2. Sample e-Learning Process for a 4-week introductory course

Agent-oriented e-Learning Model

Modeling is a technique for formulating and expressing problems and ideas (Wilcox et al.). An idea can be refined and clarified with the support of a model. In addition, people can communicate their ideas using common terminology, and common notation. E-Learning Process is a complex process that involves a lot of working components and knowledge exchanges. Agent-based modeling is the best approach to model it.

The major components in the Agent-oriented e-Learning Model are identified, as follows:

(1) **A set of related learning tasks** which are executed according to a predefined rule in order to achieve a particular pedagogical goal.

(2) **Participants** who have a particular role within the learning process. For examples, student and instructor.

(3) **Learning resources** which are entities that support the pedagogical goal like teaching note and assignment handout.

(4) **System functions** that support task accomplishment or social interactions.

Based on the typical steps to form the e-Learning Process (Figure 1), the first model is derived.

The adaptive feature has been incorporated to the model then. In our Agent-based e-Learning Process Model (Figure 3), each agent is assigned to deal with a specific type of problems and knowledge. The workloads are evenly distributed. In such a way, the model can run more robust.

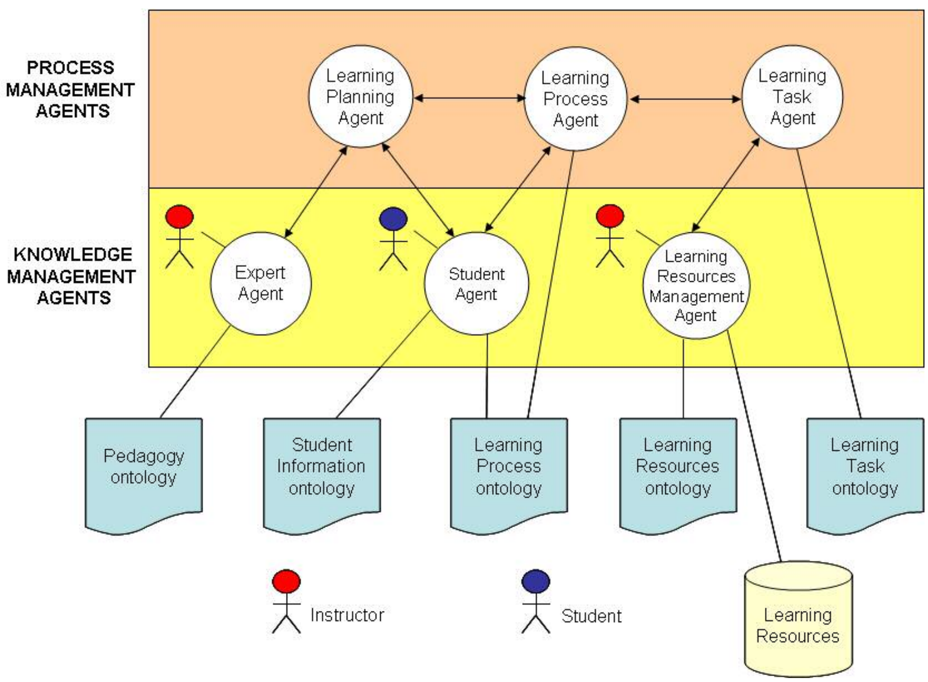

Figure 3. Agent-oriented e-Learning Process Model

Model Development

Basically, the e-Learning related knowledge is stored and manipulated using an ontology language, OWL DL (Web Ontology Language with Description Logic), which is understandable to both human and computer applications. Agents used in the model are active intelligent agents. According to the specification, they can collaborate with each other using a common language, and have reasoning and learning ability. In addition, they can manipulate knowledge related to their duty, but not all. In this section, an introduction about the use of ontologies and agents in the model is covered.

Ontology

An ontology is an explicit specification used to present a conceptualization precisely (Hwang, 2006); and according to Gruber (1993), Ontology is a formal specification of the terms in the domain and relations among them. Ontologies are widely adopted in most knowledge-base system development projects as it can express the knowledge definitely. W3C defined three sublanguages to support the OWL language. They are OWL Lite, OWL DL ("Description Logic"), and OWL Full respectively. In terms of the expressiveness, computational power and decidability, OWL DL is used. The knowledge presented in OWL language can be shared among people or software agent. In addition, this approach enables reuse of the domain knowledge.

In our model, five useful ontologies are present:

(1) **Pedagogy ontology:** This presents the means to learn (or to teach) by course. Before a course begins, instructor should set up the learning goal(s) for the course and then figure out some means to achieve these goals.

(2) **Student Information ontology:** This presents the individual student profile. Student profile is a key element to the adaptive learning process. Each student must have a distinct profile that shows the learning status and learning needs. Student Information ontology has two main sub-classes, static information and dynamic information. Static information includes personal information like student ID (the unique identifier), student name, age, education status and learning style (e.g. learn more efficient with multimedia mode or with text mode) etc, whereas dynamic information includes last learning date and last learned task etc. Dynamic information is updated more often than the static one.

(3) **Learning Process ontology:** This presents the weekly learning topic(s) by course. Like the pedagogy ontology, instructor should determine all weekly topics before the course begins.

(4) **Learning Task ontology:** This presents all available learning tasks to each course, but only a subset of them would be picked by instructor due to different reasons, e.g. level of topic, and teaching hours.

(5) **Learning Resources ontology:** This keeps track of the availability of all learning resources stored in the database. For any update in the physical resources repository, the facts stored in the ontology are then updated using offline batch mode accordingly. Most agent-based systems do not allow instant update to the ontologies, except the JessAgent. However, the offline batch approach is still being adopted in this model to reduce the operational risks.

Apart from the above ontologies, one Knowledge base is available in the model. This is the repository for all available learning resources, such as teaching slides, and exercise. It is well organized by course code. The access right to the above ontologies and Knowledge base will be covered in next part.

Intelligent Agents

In the model (Figure 3), six intelligent agents are shown. The roles, responsibility and knowledge capabilities are as follows:

(1) **Learning Planning Agent**: This agent has no authority to access any ontology. However, it acts as a planner/coordinator in the adaptive process creation. It can interact with three agents. They are Student Agent, Expert Agent, and Learning Process Agent.

(2) **Learning Process Agent:** This agent has full access rights to the Learning Process ontology. It acts as an assistant to the Learning Planning Agent in the adaptive process creation and plays a major role in the adaptive learning process. It can interact with three agents. They are Learning Planning Agent, Learning Task Agent, and Student Agent.

(3) **Learning Task Agent:** This agent also has full access rights to the Learning Task ontology. It acts as an assistant to the Learning Process Agent in the adaptive process creation and plays a supportive role in the adaptive learning process. It can interact with two agents. They are Learning Process Agent, and Learning Resources Management Agent.

(4) **Expert Agent:** This agent can fully access the Pedagogy ontology. Under normal situation, instructors can add, change, or delete the pedagogical rules on the ontology via a user interface. All these requests are handled by the Expert Agent. It can interact with Learning Planning agent for the adaptive process creation.

(5) **Student Agent:** This agent can fully access the Student Information ontology. Students can add, or change his own profile in the ontology via a user interface. All these requests are handled by the Student Agent. It can interact with two agents, Learning Planning Agent and Learning Process Agent, for both adaptive process creation and adaptive learning process respectively.

(6) **Learning Resources Management Agent:** This agent can fully access the Learning Resources ontology and the physical Learning Resources Knowledge base respectively. Instructors can add, change, or delete the learning resources from the KB via a user interface. It can interact with Learning Task Agent for both adaptive process creation and adaptive learning process respectively.

There are two extra monitoring agents in the framework, even though they are not present in Figure 3. They are responsible to poll other agents (except itself) one by one within a defined interval to check their healthiness. Both are able to restart the problematic agents when required, and all transactions done by the agents are logged in a log file for contingency purpose.

Architecture

This Multi-agent framework is implemented with the JADE framework. JADE is a FIPA-compliant agent platform. Each JADE run-time environment is called a container. A group of containers makes up a platform. Multiple containers can run simultaneously. Agents from different containers can still communicate with each other by using a method called Agent Migration. Before the agents can communicate with each other on the same platform or remote platform, they have to register in the platform directory. Each agent belongs to a container with FIPA specific Agent Management Service (AMS), Directory Facilitator (DF), and Agent Communication Channel (ACC) respectively.

Jess is a rule engine that is written in Java and is compatible with JADE. Jess and JADE work together to enable reasoning functions in the framework. Apart from this, all facts are stored in forms of ontology language and Jess rules which are inter-convertible. Instructor is a domain expert, but not a technical people, so a user-friendly ontology editor called Protégé-OWL can be used. Instructor can construct the ontologies using the editor instead of using computer scripting. However, the system administer is still allowed to use of a JessTab extension in Protégé to input the Jess rule via the Protégé editor. JessTab is a bridge between Protégé and Jess, which means that Jess rules can match on Protégé instances. The JessTab interface is formed after the Jess-to-Java coupling. With this coupling, it is possible to prototype the system in Java, Jess, and Protégé. By using the command *"mapinstance <instance-name>|<instance-address> [nonreactive | reactive]*", a specific Protégé instance can directly map to a Jess fact simultaneously.

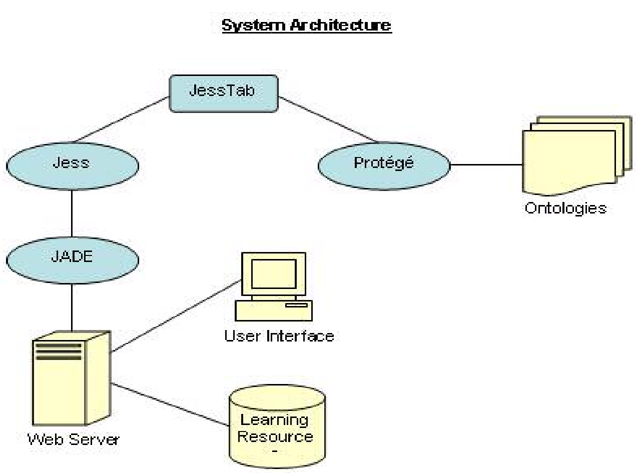

Figure 4. System Architecture

Once the platform is initiated, the first registered container is called Main-container. The main-container provides the following functions to facilitate communication among agents:

- **Agent Management Service (AMS):** It manages the life cycle (creation, deletion, and authentication) of agents and provides a white pages directory service for the agents on the platform.
- **Directory Facilitator (DF):** It offers yellow page service to lookup an agent for specific service.
- **Agent Communication Channel (ACC):** It provides the foundations for agents to communicate. **Agent Communication Language (ACL)** is a common message format used for exchanging information between agents. ACC facilitates exchange of ACL between agents. The ACL message format used in this e-Learning framework is defined in FIPA. All agents have to communicate using this messaging scheme.

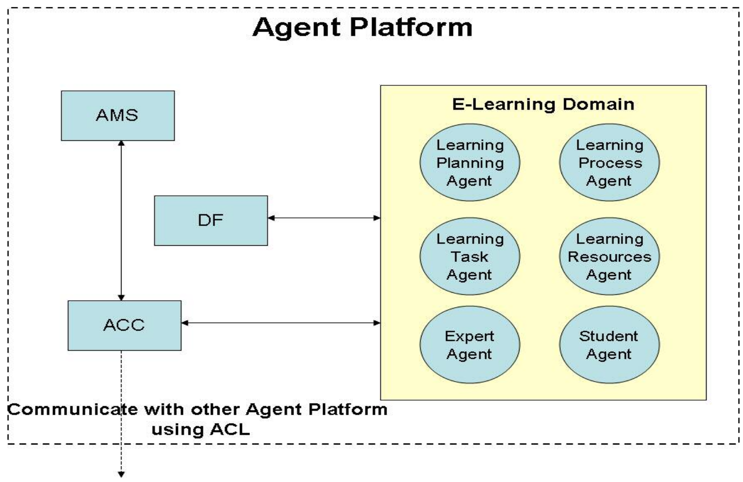

Figure. 5. Agent Architecture

Message Processing

ACL message is the only means for information exchanges among agents. Two types of messages are used, REQUEST and CONFIRM. Agents use REQUEST message to request for services and use CONFIRM message to respond to the REQUEST message they have received. They are not able to process other message types. If they receive any message with a type other than these two, they would simply return a NOT-UNDERSTOOD message to the sender agent.

Once an agent has received any ACL message from other agent, it starts execution of the message processing logic (Figure 6). It first checks the message type. If the type is not valid, then the agent is simply return a NOT-UNDERSTOOD message to the sender.

In case an agent needs any support from other agent, it initiates a REQUEST message with the request in the message context. On receiving any REQUEST message, this means that one agent is looking for service(s) that the receiver agent can offer. The receiver agent first transforms the message content into ontology format (Figure 7) and then interprets the request using its rule-based reasoning engine by calling the Jess using a Jess's Java API. The requested information is then enclosed in the returning CONFIRM message. For any request that the receiver agent cannot be solved solely, the receiver agent needs to initiate a new REQUEST message to the agent that can help according to the DF or simple rules in its working memory.

On receiving CONFIRM message, the receiver agent knows its request has already been handled by the sender agent and the required information is enclosed in the message content. The receiver agent then stores the message content in ontology format and updates its internal status accordingly.

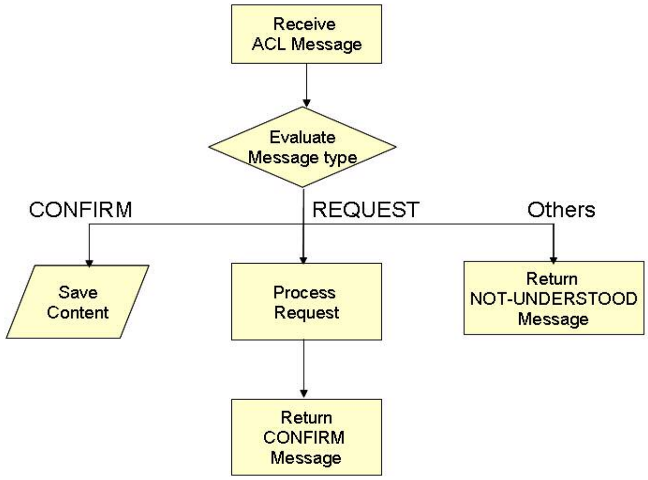

Figure 6. ACL Message Processing Logic

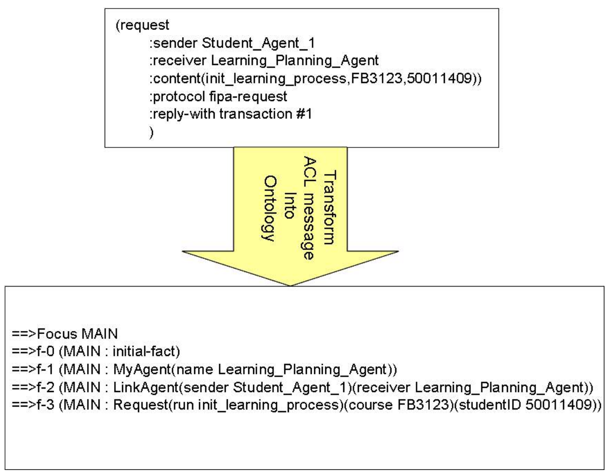

Figure 7. Transformation of ACL message into ontology

Case Scenario

A case analysis for the course of FB3123 Database System Concepts is used for proofing the development.

One learning goal for the course FB3123 is to let the students gain practical experience in manipulating database systems. Practical experience is one type of applied knowledge which can gain through hand-on practice. Therefore, doing exercise is one mean to achieve the goal. The instructor provides suggested solutions to the student. In general, instructor presents the lecture by using electronic presentation slides. As a short summary, the means to learn the course include presenting the lecture using presentation slides, providing hand-on exercise, and solution respectively.

Based on the pedagogical goals of the course FB3123, instructor needs to input the pedagogical ontology (Figure 8), and learning Process ontology (Figure 9) as well as selects tasks for the course from the Learning Task ontology (Figure 10) before the course begins. In order to make sure that the learning resources required by the learning task are available, Learning Task Agent would check the availability with the Learning Resources Management Agent internally.

```
<rdf:RDF>
 ...
<owl:Class rdf:ID="FB3123">
   <rdfs:subClassOf>
     <owl:Class rdf:ID="FB3123_Pedagogy"/>
  <owl:Class rdf:ID="text">
    </rdfs:subClassOf>
  </owl:Class>
….
<owl:Class rdf:ID="multimedia">
    <rdfs:subClassOf rdf:resource="#presentationSlide"/>
  </owl:Class>
    <rdfs:subClassOf rdf:resource="#presentationSlide"/>
  </owl:Class>
…
  <owl:Class rdf:about="#exercise">
    <rdfs:subClassOf rdf:resource="#FB3123_Pedagogy"/>
  </owl:Class>
…
</rdf:RDF>
```

Figure 8. Pedagogy ontology

Ontology can be used to present the knowledge in a more flexible way than database or indexed file. For instances, instructor may cover a dynamic number of topics in a week. Ontology can present this kind of dynamic situation without paying extra effort as Figure 9.

```
<?xml version="1.0"?>
<rdf:RDF
  ...
 <owl:Class rdf:ID="FB3123_LearningProcess">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >FB3123 Database Systems Concept – Semester A, 2006/07</rdfs:comment>
 </owl:Class>
...
 <owl:Class rdf:ID="#topic_2">
   <rdfs:subClassOf rdf:resource="#week_2"/>
 </owl:Class>
 <owl:Class rdf:ID="#topic_3">
   <rdfs:subClassOf rdf:resource="#week_2"/>
 </owl:Class>
 <owl:Class rdf:ID="#week2">
   <rdfs:subClassOf rdf:resource="#FB3123_LearningProcess"/>
 </owl:Class>
…
```

Figure 9. Learning Process ontology

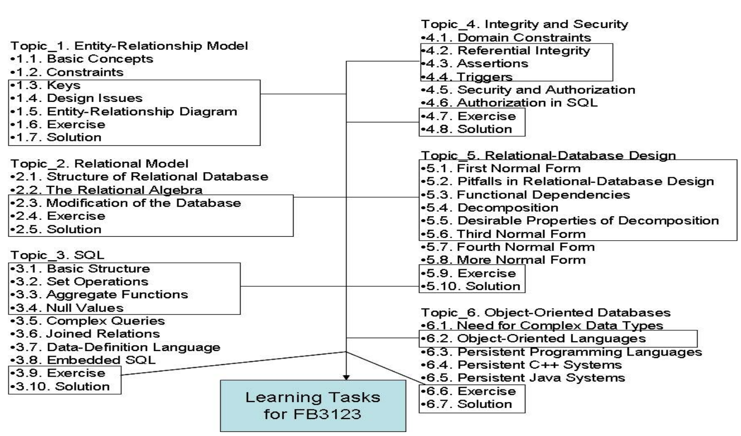

Figure. 10. Selected learning tasks for FB3123

When all ontologies are set, the course is set to be available to students. Once student has login to the framework, a new Student Agent is created automatically. It initiates the adaptive e-Learning Process creation by sending a request with the course code (i.e. FB3123) to the Learning Planning Agent. Learning Planning Agent acts as a coordinator for the e-learning process creation. It needs to gather the Learning Process ontology, Learning Task ontology and Learning Resources ontology which can be accessed by the Learning Process Agent, Learning Task Agent and the Learning Resources Management Agent respectively. According to the DF, these four Agents can cooperate and gather the ontologies by passing ACL messages.

Apart from these three ontologies, the Learning Planning Agent has to gather the pedagogy ontology from the Expert Agent and the Student Information ontology from the Student Agent as well. The Learning Planning Agent is capable to generate a general e-learning process for the course with the Pedagogy ontology, Learning Process ontology, Learning Task ontology, and Learning Resources ontology. It then extracts and applies the adaptive elements, e.g. learning style (e.g. multimedia mode or text mode) from the Student Information ontology to the general e-learning process and derives an adaptive e-learning process into the working memory. Students can start learning.

Learning Process

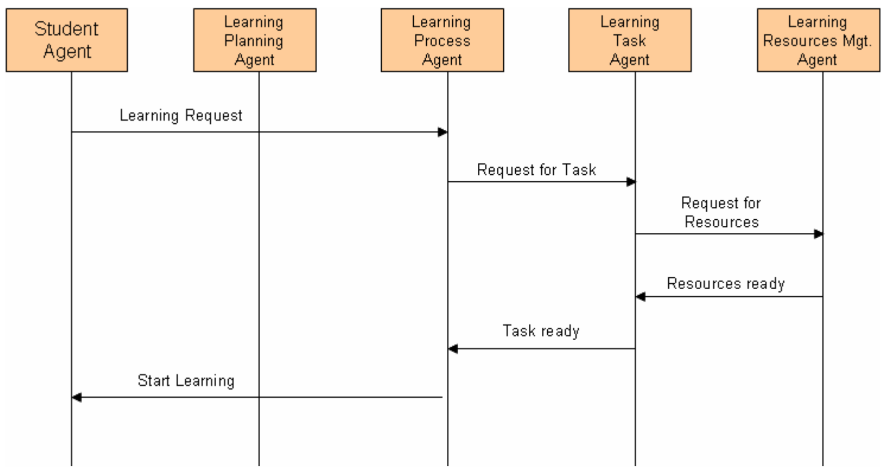

Figure 11. Workflow for Learning Process

After the adaptive learning process is ready in the working memory, the Student Agent is able to send a Learning REQUEST with the course code (e.g. FB3123) to the Learning Process Agent to initiate the learning process.

When the Learning Process Agent receives the Learning REQUEST, it begins to work. It first transforms the message from ACL format to ontology format (Figure 7) which can be interpreted by the rule possessed by the Learning Process Agent. Then the reasoning result would instruct the agent to retrieve the next learning task name and the required learning resources name from the adaptive learning process in the working memory, and send a REQUEST for task with the next learning task name plus the required learning resources name to the Learning Task Agent. The purpose is to make the required learning resources available to the student.

When the Learning Task Agent receives the REQUEST for task, it begins to work. It first transforms the message from ACL format to ontology format and then interpreted it by the rule possessed by the Learning Task Agent. Then the reasoning result instructs the agent to send another REQUEST for resource with the required resources name to the Learning Resources Management Agent which can access the Learning Resources KB.

When the Learning Resources Management Agent receives the REQUEST for resources, it retrieves the required resources directly from the Learning Resources KB and makes it available to the Student through the user interface and then returns an acknowledgement to the Learning Task Agent. The Learning Task Agent further returns an acknowledgement to the Learning Process Agent.

Once the student has completed the current task, the Learning Process Agent would make suggestions on the preceding learning task according to the adaptive learning process in the working memory. However, student can still override the suggestion. Simultaneously, the Learning Process Agent keeps waiting for any other learning request from the Student Agent.

Evaluation and Discussion

In this section, there is a comparison between the features of the proposed agent-based process-oriented e-Learning system and that of the traditional e-Learning system without such support. All in all, the Agent-based Process-oriented e-Learning System is more flexible and more compatible with the pedagogical theory rather than the traditional ones.

Features	**Traditional e-Learning systems without agent and process management support**	**Agent-based Process-oriented e-Learning System**
Learning task management	Like Blackboard and WebCT, it has no task management feature. It just provides content management support.	The tasks have been arranged in sequence. Students can learn in process basis that has been defined by the instructors. There is a higher chance to achieve the predefined pedagogical goal.
Manipulation of pedagogical rules	No linkage with the pedagogical rules. Inconsistency between the pedagogical rules and the course content may happen.	The learning process is generated with reference to the pedagogical rules. Instructor can change the pedagogical rules when the course is running as well. The learning process can be regenerated without human intervention.
Operational constraints	Only simple logic can be supported. It cannot deal with complicated logic.	Rules can be used to exercise control of tasks, and prohibit unauthorized operations (Wang and Wang, 2006). With rule-based reasoning, complex logic can be supported.
Reaction to change	They cannot react to any unknown situation.	Any changes in ontologies can be handled dynamically without paying extra effort.
Customization to individuals	Most of them are incapable to support personalization.	Agent technology with rule-based reasoning function enables dynamic personalization feature.
Support of interactive learning tasks	They are quite passive. All the requests are initiated by students.	The learning is more interactive. After student has completed one task, then the agent suggests next task proactively.

Conclusion

There is strong evidence that students learn through learning process instead of individual tasks. Like the problem solving heuristic process, problem-solving skills can be acquired and strengthen through a learning process with a set of inter-related learning tasks. Students learn according to the predefined workflow instead of taking the tasks individually without a sense. It shows that the process-oriented approach can achieve better learning outcomes. However, almost all existing e-Learning tools are resource-oriented instead of process-oriented. It definitely affects the learning outcomes. In addition, there exists diversity in the student profiles, so the need for a generic e-Learning Process Model with adaptive feature is very prominent.

The proposed agent-based e-Learning Process Model enables: (1) student to learn adaptively according to the pedagogical rules, his own profile and his learning progress, (2) instructor to enhance the pedagogical rules and add learning resources in such a dynamic learning environment, (3) e-Learning tool designer to refer to when they intend to design an e-Learning system which can really fit the pedagogical principles. Based on this framework, a prototype is under implementation. The details together with other relevant issues, like the reuse of learning objects in e-Learning will be discussed in the next work.

Reference

ADL, SCORM-3rd Edition Overview, 2004, retrieved March 1, 2007, from http://www.adlnet.org.

Aroyo, L., Kommers, P., "Intelligent agents for educational computer-aided systems", Journal of Interactive Learning Research, 10(3/4), 235-242, 1999.

Hawryszkiewycz, I.T., "A Framework of Integrating Learning into Business Processes", Conferences in Research and Practice in Information Technology, 46, 2005.

Hwang W. J., "Design and Implementation of Multimedia Service Management Agent on Home Networks Environment", International Journal of Computer Science and Network Security, 6(7B), 2006.

Krulik, S. & Rudnick, J.A., "The New Sourcebook for Teaching Reasoning an Problem Solving in Junior and Senior High School", Boston: Allyn and Bacon, 1996.

Ma. C.M., "Students' Experiences of Understanding Nonroutine Problems: A Phenomenographic Analysis", The Asia-Pacific Education Researcher, 13(1), 87-133, 2004.

Marjanovic, O., "Towards A Web-Based Handbook of Generic, Process-Oriented Learning Designs", Educational Technology and Society, 8(2), 2005, 66-82.

Proctor, C., "Teacher Expectations: A Model for School Improvement", The Elementary School Journal, 469-481, 1984.

Wang, M. H., and Wang, H. Q., "From Process Logic to Business Logic – A Cognitive Approach to Business Process Management", Information and Management, 43, 179-193, 2006.

Wang, M. H., Wang, H. Q., and Xu, Dongming, "Intelligent Workflow Monitoring with Agent Technology", Knowledge-Based Systems, 18(6), 257-266, 2005.

Weske. M., Van Der Aalst. W.M.P., and Verbeek, H.M.W., "Advances in Business Process Management", Data and Knowledge Engineering, 50, 2004, 1-8.

Wilcox, P., Dexter, H., and Petch, J., "A Foundation for Modelling e-Learning Processes", The e-Learning Research Centre, The University of Machester.

Wooldridge, M.J., and Jennings, N.R., "Intelligent Agents: Theory and Practice", The Knowledge Engineering Review, 10(2), 1995, 115-152.

Xu, D. M., and Wang, H. Q., "Intelligent Agent Supported Personalization for Virtual Learning Environments", Decision Support Systems, 42(2), 2006, 825-843.