

Intelligent Instructional Resource Planning System for an Enterprise eLearning Management System

Thaveesak Ganjanasuwan¹, Charan Sanrach²

Department of Computer Education
King Mongkut's Institute of Technology North Bangkok
Bangkok, Thailand
eddy_tg@hotmail.com¹, jsr@kmitnb.ac.th²

Abstract

This paper studies the development of Intelligent Instructional Resource Planning System (II-RPS) for an Enterprise Learning Management System. In the context, II-RPS Implementation is to introduce, customize and exploit in the form of a prototype of an integrated multi-agent for automation of courseware production planning. Users of these systems will include course directors, authors and co-authors. It has been developed within the model and architecture of multi-agent instructional resource planning. This new system is not only used to promote the automatic learning process through systematic framework of instructional resource planning, but also to increase the performance of courseware production and to highly benefit the pioneer who uses this approach.

1. Introduction

Effective learning management has been achieved through usage of learning technology that is based on computer management learning (CML) for student/learner-centered, including learning management systems (LMS), learning content management systems (LCMS), knowledge management systems (KMS), intelligent tutoring systems (ITS), electronic

performance support systems (EPSS) and other related support systems, all of which promote an automatic learning process. Nevertheless, one crucial component is missing-a systematic framework of instructional resource planning [1] that relates to the report of EuroPACE, 2002 [2] and from experience in research and development of web-based training system [3].

It is generally accepted that instructional resource planning is a larger and more complex domain problem. Developers are now building enterprise-wide and global applications that must operate across corporations and continents. More and more corporations need to integrate their information systems with those of their planners and developers. Developing an application for these existing and emerging application domains requires powerful new methods and techniques for conceptualizing and implementing software systems.

One way to obtain an efficient instructional resource planning is through usage of artificial intelligence, which imitates human behavior and the human thinking process including planning, monitoring and controlling. Multi-agent provides a powerful solving-problem paradigm that is well suited to developing complex enterprise applications. Nevertheless, it is also a new class of software that acts on behalf of the user to automate a complex task,

communication and co-operate with other agents to solve complex problems.

Accordingly, a II-RPS architecture has been set up with adoption of multi-agent for instructional resource planning model such that the instructional resource planning becomes “intelligent.” The following two objectives are to develop II-RPS and to evaluate the developed model and system by experts and users.

2. Related Research Topic

“Instructional Planning is the process of mapping out a global sequence of instructional goals and actions that provides consistency, coherence and continuity throughout an instructional session” [4].

“Instructional Planning, in the context of machine planning for ITSs, is the process of configuring partial orderings of instructional operations that when executed are likely to result in a student achieving selected instructional objectives” [5].

In the context of a computer-based learning environment, an instructional planning can be defined as any form of interaction between the student and the instructional material or the computer-mediated interaction between collaborating students or between the student and a teacher. The instructional component of a computer-based learning environment is responsible for determining what to do next at each point in an instructional interaction, hence, it is in control of the system's behavior. Instructional planning, the application of AI planning techniques in the domain of instruction is a major venture that is related to this research topic and will be adopted and described below.

The primary research study for the correlations of the two research papers namely “Instructional planning in an intelligent tutoring System: Combining Global Lesson Plans with Local Discourse Control” [6] and “Using Case-Based Reasoning in

Instructional Planning (CBIP): Towards a Hybrid Self-improving Instructional Planner” [7] methodically studied the results of different approaches, but the same limitations of cooperative and instructional resource planning for allocation. Some characteristics of intelligent agent system need to increase more efficiency.

The last research study for the correlation of two-research papers namely “CAMPS: A Constraint-Based Architecture for Multi-agent Planning and Scheduling” [8] and “Multi-Agent Planning and Scheduling Environment for Enhanced Spacecraft Autonomy” [9] systematically explored. It is not only that they are different approaches with the same limitations, but the technique of the research paper [9] is also more efficient for meaningful and useful resource planning allocation.

The optimistic idea is to carry out some distinguished features of [6], [7], [8], and [9] by using a multi-agent for instructional planning. It is not only distributed instructional planning, but cooperatively using the protocol of Knowledge Query Manipulate Language (KQML), instructional resource planning, real-time monitoring and integrate cross platform techniques in order to achieve the highest levels of courseware production with more effectiveness and efficiency.

3. Development of II-RPS

System Development Life Cycle (SDLC) in reference to adaptive waterfall process as adopted and described by Dennis and Wixom [10]. The major phase consists of Planning, Analysis, Design and Implementation. However, this process must be related and based on the existing model and architecture of multi-agent instructional planning.

3.1 Multi-Agent Instructional Resource Planning Model

3.2 II-RPS Architecture

A model of Multi-Agent instructional resource planning for an enterprise eLearning management system [11] can be purposely created as shown in Figure 1. The proposed model uses the “System’s Theory Model”, both in its planning networking and its resource as well as its scheduling, as an input to the “Agent Process Model” of which processing steps are detectors, parse input, determine action, invoke action and effectors. Then the outcome of the courseware processing is compared to the target, which is the amount of processing time. If any task is not completed within a specified amount of time, there will be a feedback through the effector to the re-planning section, as well as the other parts again and again until courseware outcome (performance) meets the target. The model consists of three components as shown in the following processes: “Input” consists of task/activities, instructional resources and scheduling. “Process” consists of detectors, parse input, determine action, invoke action, effectors, state and replanning. “Output” is the end result of courseware is reasonably compared to achieve the constant goal (Time).

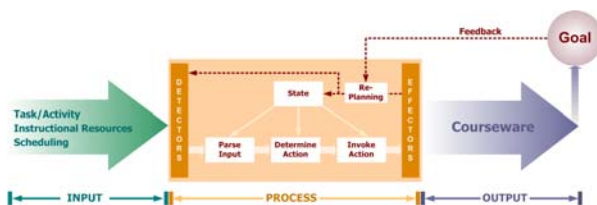


Figure 1: Multi-Agent instructional Resource Planning Model

The architecture of II-RPS [11] can be purposely created as shown in Figure 2. The II-RPS is developed with Java and XML and integrated the LCMS via agent engine that enables planning the courseware production, executing and effectively controlling. The planning process will automatically alert the user’s responsibility within the workflow and then distributes application through instance agent online system. The teacher/author agent makes an approval request for a courseware production plan and then distributes the task to the co-author agent that can be in real-time monitoring, while the course director agent accepts to approve the plan. The co-author will produce the courseware from the task assignment via HTTP, JDBC database connection and the instruments in the LCMS. If the plan fails to achieve the goal (time), the course director is able to re-plan.

The strength of the system architecture not only interacts between the course planner and a predetermined goal regarding system operation, goal setting and goal achieving, but also enhances effectiveness in the systemic planning of courseware production. That entails planning, plan proposing, plan approval, plan segmentation and plan execution, including a route for the planner to monitor the entire process through “Agent Engine”, which enables the planner to promptly carry out a modification if necessary. The system also uses “Planning Wizard” function which is user-friendly and very cost efficient in the long run, for example, it can automatically update, analyze, evaluate and report data.

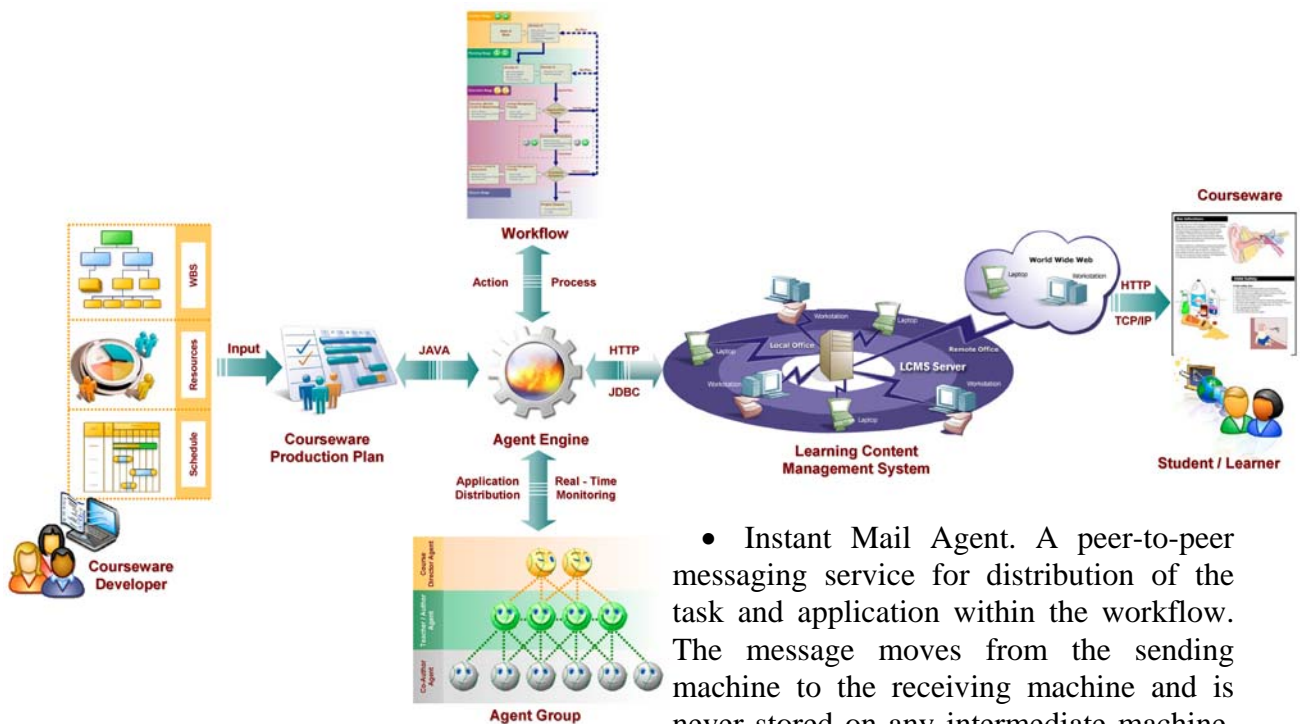


Figure 2: The Architecture of II-RPS

3.3 II-RPS Functionality

The requirements will be classified into a list of system functional specifications that satisfy those requirements as shown in Figure 3. II-RPS functionality can be divided into three modules as follows;

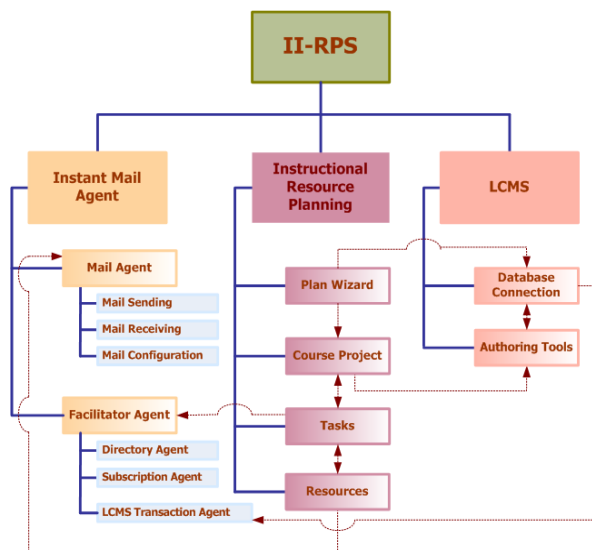


Figure 3: II-RPS Functionality

- Instant Mail Agent. A peer-to-peer messaging service for distribution of the task and application within the workflow. The message moves from the sending machine to the receiving machine and is never stored on any intermediate machine. An agent can communicate with any other agent that it knows about and will accept its message by using KQML via TCP/IP. A facilitator agent is the control panel that provides to manage communication between the agent and can be viewed as providing real-time agent name services and access services to instant mail agent, including LCMS transaction monitoring.

- Instructional Resource Planning. A process of mapping out a sequence of instructional goals and actions that provides plan wizard, course project, task and resource throughout the LCMS in an author mode.

- LCMS. A system for collaborative development of learning content with built-in database connection and authoring tools.

3.4 II-RPS Development Process

Developing an II-RPS consists of identifying the roles and function of various agents and then specifying each agent's behavior. An agent programming language is a Java code used to specify the behavior of the agent for any given situation. This

developing process can be described as shown in Figure 4. The three-step process of writing code, compiling code and executing code. Writing a java code creates an instant mail agent program and is used to organize programming and debugging agent/agencies. Integrated java code can be incorporated with the instruction resource planning and LCMS via a JDBC database connection and HTTP to link the authoring tools for courseware production before compiling the code. The final step of executing the program is to interpret into the run-time agent engine. The agent engine is an execution mechanism that interprets the agent programs and performs the action specified in the user interface and agent action libraries.

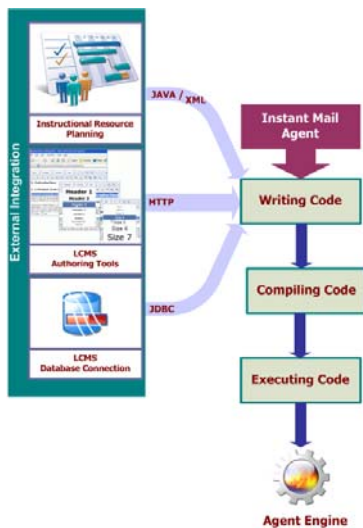


Figure 4: II-RPS Development Process

3.5 II-RPSagent Based Classes

II-RPSagent is the base class that defines a common programming interface and behavior for all the agents in the architecture of II-RPS. II-RPSagent class uses several helper classes as shown in the Figure 5.

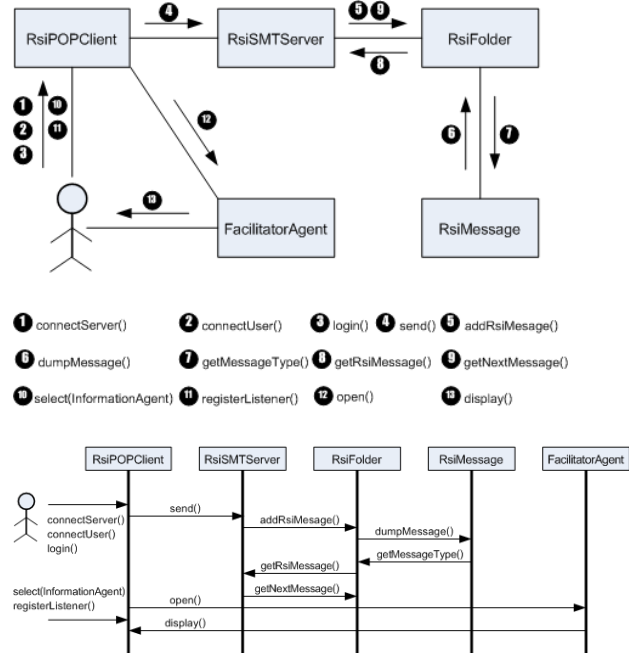


Figure 5: II-RPSagent Collaboration and Sequence UML Diagram

Figure 5, II-RPSagent collaboration and Sequence UML diagram shows the relationship between the classes that make up a functional II-RPS agent instance. II-RPSagent class diagram can be classified into five wrapped classes as follows;

- RsiPOPClient class provides access to a POP3 server. This class does the entire interaction with the POP3 server. It handles connecting, sending user name and password, as well as downloading mail. It downloads mail as a folder, not message by message. (The folder is of type RsiFolder).
- RsiSMTPServer class provides functionality to use an SMTP mail server. It handles all responsibilities relating to the sending of email. It needs to be constructed with the name of an IPAddress of an SMTP server that is willing to handle the client's mail.
- The RsiMessage class is the data class for an email message. It contains two import fields, a “head” and a “body”. There is a “dumpMessage” that prints a complete version of the message to a string.
- The RsiFolder class is a container class for RsiMessage objects. It contains a vector

of messages that can be manipulated in several ways. The developer can iterate through a folder by going “nextMessage” as well as retrieving them based upon an index. It has a “dumpMessages” method that will dump all messages to a string (useful for development).

- The FacilitatorAgent class provides to manage communication between an agent that can be viewed as providing a real-time agent directory and subscription services to an instant mail agent, including LCMS transaction monitoring.

3.6 II-PRS Demonstration Overview

The purpose of II-RPS demonstration is to illustrate the use of multi-agent in instructional resource planning for an enterprise eLearning management system as shown below.

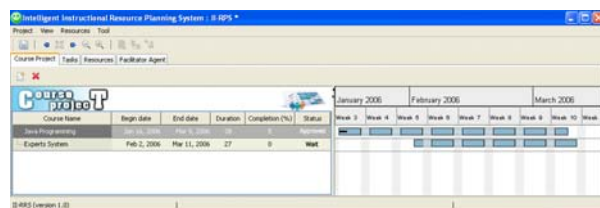


Figure 6: Course Project

Figure 6 illustrates “Course Project” screenshot, the course author will do the course planning through the planning wizard by entering into the mode of the subject the author wants to work on. Then, the author will specify the parameters which are ‘course name’, ‘begin date’, ‘end date’, ‘duration’, ‘completion%’ and ‘status.’ The default value of ‘status’ is ‘wait’ as the author needs to wait for permission from the course director. The default value of ‘begin date’ and ‘end date’ are varied by the time specified in ‘tasks.’ ‘Gantt chart’ will show the progress of an approved course planning so as to facilitate the monitoring and controlling of the course planning.

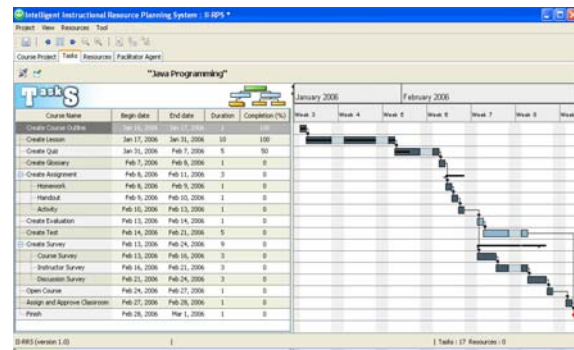


Figure 7: Tasks

Figure 7 illustrates “Tasks” screenshot in different phases. The ‘tasks’ must work in harmony with the author’s task(s) in the LCMS system including ‘create course outline’, ‘create lesson’, ‘create quiz’, ‘create glossary’, ‘create homework’, ‘create handout’, ‘create activity’, ‘create evaluation’, ‘create test’, ‘create course survey’, ‘create instructor survey’, ‘create discussion survey’, ‘open course’, and ‘assign and approve classroom,’ by connecting to each of the tasks’ page. The tasks are fixed, however. The course author will not be able to add more tasks or delete some of the tasks from the existing list. But, again, the parameters in the ‘tasks’ can be adjusted. ‘Gantt chart’ will show the time line of the planning and the progress of the planning according to the parameter of ‘completion%.’ ‘Gantt chart’ will also show ‘critical path method’ which reports the longest-time path considered to be the ‘critical path’ so as to facilitate the course director and the course author to well manage the planning within the time line.

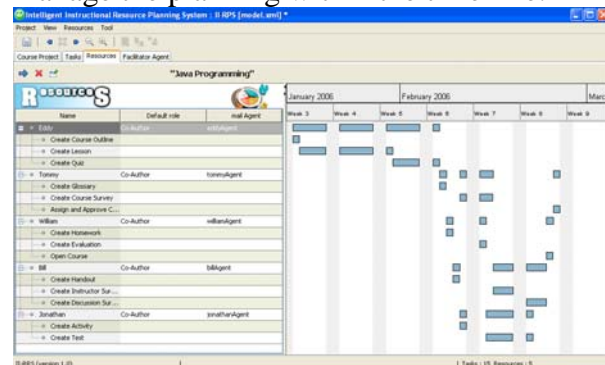


Figure 8: Resources

Figure 8 illustrates “Resources” screenshot. The course author can work on the human resource for each task on his own, by specifying the name, default role and email, etc. Once the plan is approved, the system can notify relevant personnel via the ‘instant mail agent.’ ‘Gantt chart’ will show the ‘slot time’ of the personnel. If some of the personnel’s working hours exceed the limit, the system will make a notification.

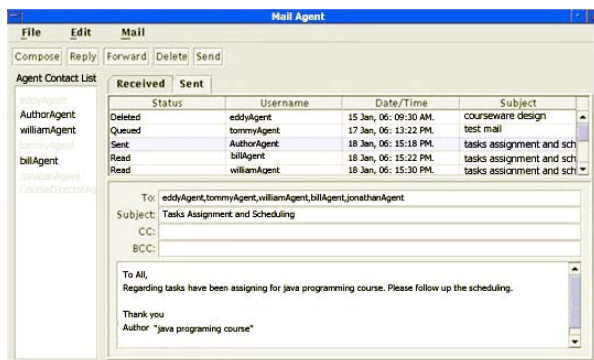


Figure 9: Mail Agent

Figure 9 illustrates “Mail Agent” screenshot, which is part of the ‘instant mail agent.’ A simple of panel is divided into three major segments. These are the agent contact list, the mail status display, and the mail composition/viewing area. Mail Agent also provides a user interface for reading, composing, sending and receiving messages. All client agents communicate with the facilitator agent.

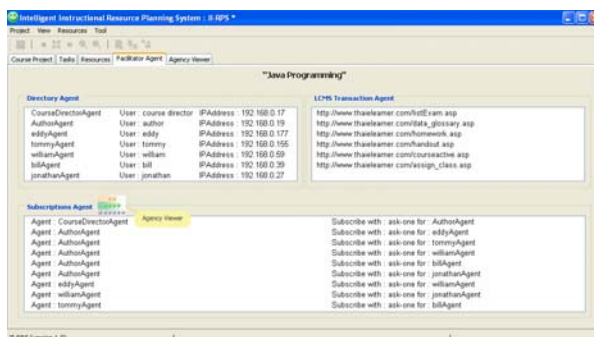


Figure 10: Facilitator Agent

Figure 10 illustrates “Facilitator Agent” screenshot, which is part of the ‘instant mail agent.’ The course director and the course author can do real-time monitoring through the ‘console’ including ‘directory agent’ for a search of the location of an agent, ‘subscriptions agent’ for monitoring the communication between agents through the ‘KQML protocol’, and ‘LCMS transaction’ for tracing back previously executed job(s) in each of the ‘tasks’ in the database. All of these make the monitoring even more efficient.

Figure 11 illustrates “Agency Viewer” screenshot that shows the flow of messages between agents and allows the user to examine the contents of the messages in real-time. There are three different agent types involved, course director agent, author agent and co-author agent. Each kind of agent has a different role and different capabilities. The course director agent is instantly responsible for interacting with the author agent, while the author agent promptly stimulates sending and receiving the mailing task from any number of co-author agent. The co-author agent also communicates and subscribes all operational activities with each other using the KQML.

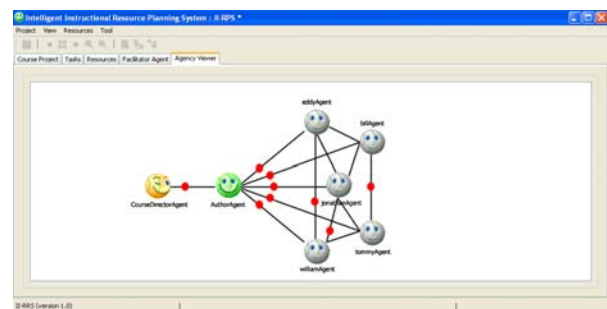


Figure 11: Agency Viewer

3.7 Courseware Verification

One of the key challenges of II-PRS is the ability to test the output of the courseware. The II-RPS offers the course planner and course developer a mechanism to test, evaluate, re-plan and re-develop their

courseware through the agent engine that can interact with and interpret the courseware and model produced by the II-RPS.

The II-PRS not only allows the course planner to publish their plan in the form of XML that contains such information as the course plan manifest, instructional resource and schedule structures, but also provides the course developer the ability to create their courseware through authoring tools in LCMS. Nevertheless, the instant mail agent is used during the runtime execution /reconciliation of the courseware production planning, which allows the course planner to test the exchange messaging and real-time monitoring task agent.

3.8 II-RPS Evaluation

The initial evaluation of the II-RPS is concerned with its user-friendliness, the user's ability to create simple courseware and the user's understanding of the model's used by II-RPS. The evaluation process involved initial presentations of the functionality of the II-RPS-a demo of how it can be used. Testers were provided with a pre-defined subject matter course plan and asked to develop courseware based on the instructional resource planning. The users were satisfied with the interface provided and the ability to create and describe the subject matter area. They felt empowered by the ability to describe course lessons in a graphical manner. By allowing courseware verification, the tester felt it would increase the potential stability of the end course. However, testers felt the view of the facilitator agent was confusing as a real-time task monitoring agent and the course planning process could be made easier by using simple templates. Look and feel characteristics were inconsistent.

The second phase of evaluation will be carried out in both the Computer Education Department and the Information Technology Department at King Mongkut's Institute of Technology North Bangkok, Thailand. The

results and feedback from both evaluation phases will influence the future development of the II-RPS

4. Conclusion

This paper introduces research in the area of multi-agent, instructional resource planning and enterprise eLearning management system, which resulted in the development of the II-RPS. This paper introduces a multi-agent instructional resource planning model for building courseware based on instructional planning in a multi-agent process. This paper describes a architecture for supporting the course planner and course developer during the process of planning and creating courseware. This paper identifies and illustrates the core components required for building II-PRS. The paper concludes by outlining the initial II-RPS evaluation results.

5. References

- [1] Karl M. Kapp (2001), "Learning Requirement Planning: An Enterprise-Wide Learning Management Approach" Available online at http://www.elearningguru.com/wpapers/kapp_lrp.pdf
- [2] EPYC, "EPYC for eLearning: LCMS Learning Content Management Systems" Available online at http://www.europace.org/seminar/EPYC/seminar3/EP_LCMS_20021114.ppt, 2002.
- [3] Thaveesak Ganjanasuwan (1999), "Development of Web-based Training Systems: A Case Study of Aeronautical Radio of Thailand", Thailand: AIT.
- [4] Wasson, B. (1996), "Instructional planning and contemporary theories of learning: Is this a self-contradiction?" In P. Brna, A. Paiva & J. Self (Eds.) Proceedings of the European Conference on Artificial Intelligence in Education, 23-30, Lisbon: Colibri.

- [5] MacMillan self-improving instructional planner for intelligent, S.A. & Sleeman, D.H. "An architecture for a tutoring systems", *Computational Intelligence*, 3, 17-27, 1987
- [6] Chong Woo Woo (1991), "Instructional Planning in An Intelligent Tutoring System: Combining Global Lesson Plans with Local Discourse Control", USA.
- [7] Jon A. Elorriaga, and I. Fernandez-Castro (1998), "Using Case-Based Reasoning in Instructional Planning: Towards a Hybrid Self-improving Instructional Planner", Spain.
- [8] Kazuo Miyashita (2001), CAMPS: "A Constraint-Based Architecture for Multiagent Planning and Scheduling", Japan.
- [9] Subrata Das, Paul Gonsalves, and Raffi Krikorian (2001), "Multi-Agent Planning and Scheduling Environment for Enhanced Spacecraft Autonomy", USA.
- [10] Efraim Turban, and Jay E. Aronson (2001), "Decision Support Systems and Intelligent Systems: Sixth Edition", USA: Prentice Hall.
- [11] Thaveesak Ganjanasuwan (2006), "Multi-Agent Instructional Resource Planning Approach for an Enterprise eLearning Management System", Thailand: ICDML.