Proceedings of the IASTED International Conference
WEB-BASED EDUCATION
February 21-23, 2005, Grindelwald, Switzerland

AN AGENT FRAMEWORK FOR LEARNING SYSTEMS

I.T. Hawryszkiewycz,
Department of Information Systems
University of Technology, Sydney
igorh@it.uts.edu.au

Abstract - Personalized learning systems must allow learners to choose their learning goals and learning process. This paper describes a way for providing agent support that can assist learners to do this. The paper then proposes a framework of software agents made up of two parts. One are customizing agents that assist learners to select learning materials to satisfy learning objectives and set up a learning plan. The other are managing agents that help learners to follow a ***study program*** to progress through that material and dynamically change the process as needed. The paper describes a way to describe learning process that can be used by such agents and illustrates with a small prototype.

Keywords: Learning objects, software agents, learning management systems

1 Introduction

Learning now takes place in many environments other than formal education at teaching institutions. There are also variations in the kinds of students [1] and the method of instruction [2]. Learners in educational institutions usually follow a particular instruction program. Learners in project teams build on their knowledge through a continuous and guided process of identifying learning project goals, discussing and trying ideas and recording outcomes in their learning outputs. Competency based learning [3] is also becoming more important in practical environments as is performance improvement. The latter trends usually require more emphasis on constructivist approaches [4] in either actual project situations or within case studies where learning goals are embedded in an ill structured case study. One important aspect here is how to change the learning plan depending on learner progress and to advise learners to vary their learning plans. We propose that software agents be used to achieve this goal.

Figure 1 summarizes the general process proposed for agent supported learning. A learning objective is set. Then a learning plan together with learning materials defined. In educational institutions the learning plan is usually a set of lectures and assessments whereas we propose that such plans should be customized to specialized learner needs. The learners should then be guided through the plan. Such guidance can take many forms. One may be to direct learners to expert instructors. It may also be to identify a lack of some elementary knowledge and provide ways to build familiarity of this knowledge before continuing with the main learning plan

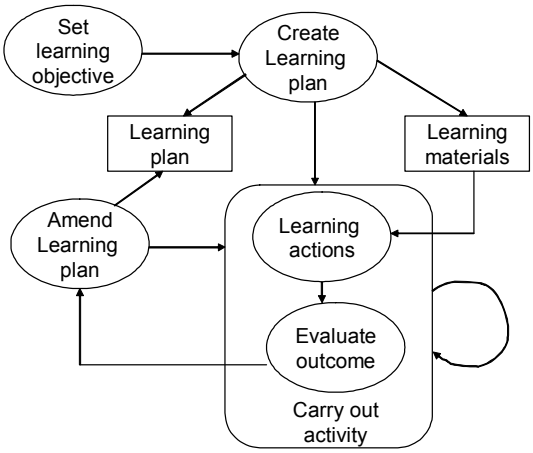

Figure 1 – General framework for self-directed learning

The challenge then is how to adapt plans to learner progress. The paper examines ways in which software agents can meet this challenge. Two classes of software objects are proposed. One class are agents that create the learning plan. The other are agents that manage the learning plan and dynamically amend the plan as needed.

One important consideration in formulating plans is to make distinction between objectivist and constructivist learning and some writers [5] suggest that most computer systems are not effective in supporting constructivist approaches. There is further emphasis on learning within the context of an extensive example, such as a case study. Student case studies are a particular example of developing competencies [3] as such competencies are better built within a larger framework usually working with others. Working with others both increases the breadth of knowledge acquired, its relevance, as well as evaluations of the application of new knowledge in a problem area and benefits from learning from peer interaction.

Flexible systems must provide a framework for storing generic learning materials and combining it into study programs. The paper develops a framework for defining such materials and provide software agent to support the learning process. The definition will use elementary objects that can be combined in flexible ways into composite objects to achieve a learning goal. The definition will also include a learning plan that defines steps to be followed in the learning process.

2 Choosing generic objects

The current research trends to the development of flexible systems are learning management systems and learning objects. Learning objects have been proposed in answer to the need to share materials across learning environments. A number of standards are now being developed for learning objects. The two most quoted standards are the Dublin core [6] and the Learning Technology Standards of the IEEE [7]. These standards usually describe objects as standard elements that include content together with ways to se it. These learning objects can be shared across the WWW in different learning environments. Many such standards assume a hierarchical structure of objects that result in large scale reusable objects such as for example entire subjects.

Many writers [8] suggest that learning objects are still a concept that needs some adjustment to a suitable implementation. Issues here include the level of granularity of such objects and frameworks to assemble such objects into learning environments that can be setup and managed by agents. We are developing a way to describe learning environments in terms of selected classes of elementary objects. In particular we separate content from method of use to provide more flexibility in constructing learning environments.

Our work has proposed a learning model that provides a basis for learning objects of fine granularity. The learning model outlined in this paper has been described earlier [9] and is shown in Figure 2. It is based on choosing learning objects to provide the flexibility to construct a variety of learning environments. The model described earlier [9] supports a finer granularity than entire subjects and extends it by proposing object classes that can be used to construct larger activities. It uses the following terminology:

Subject metadata – what is being taught. This sets a framework for discovery and is usually implemented as links within the subject metadata structure. Thus teaching for example about databases may place it within the context of businesses or applications.

Learning environment – where the learning takes places and what support is included. It includes the setting of learning goals, and the support services for a learning method. The learning goal defines what is to be achieved. Thus in a University the goal may define assessment procedures whereas in a project environment the emphasis is on project goals.

Learning activity – Defines what the learner must do to achieve the learning goal using learning objects from the subject metadata,

Learning method – defines the way that learning will take place and what learners actually do in each step of the learning activity. The chosen methods will be based on learner's cognitive preferences. Examples of methods may be reading and self-assessment, or a group case study. The methods will use any support services provided by the environment.

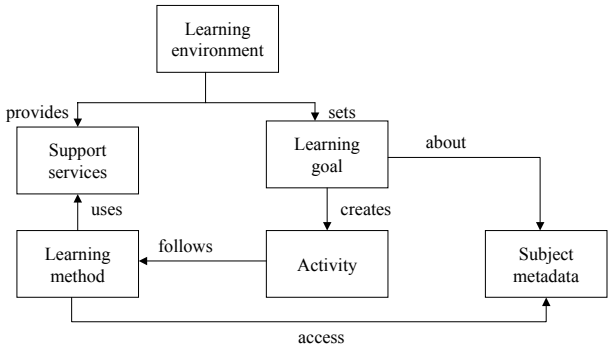

Figure 2 – Learning object classification

3 Defining Learning Environments in terms of Elementary Objects

Our work concentrates on identifying elementary objects and combining them with learning content. They also include assessment objects that can be used to dynamically change learning plans. What we need is a

model that combines the objects that make up the learning environment with agents that provide the active support.

3.1 Structure for learning

We use an XML description of structure like that used by Koper [2] to define our learning model. The definition is shown in Figure 3. We have identified a number of differences from Koper's ELM model. The differences are mainly in object parameters, which of necessity must be open and define learning goals rather than defining specific objects to enable agents to choose alternate strategies.

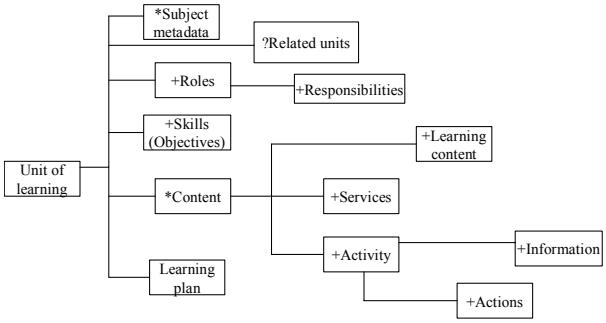

Figure 3 - Defining learning models

Figure 3 shows the potential elementary objects – subject metadata, roles, skills, content, learning plan. The idea behind the structure is that a unit of learning, which is a composite object, can be made out of any of the available elementary objects. Such elementary objects can be quite simple such as a discussion board service. This can then be combined with an object such as a report and result in a learning activity such as jointly evaluate report. The elementary object can also be a plan that brings together a number of learning activity. Thus there may be a composite unit of learning called "How to design collaborative systems". This unit of learning can use a number of different plans, as for example, a case study learning plan, or an individual study plan. A learning plan has a number of steps each of which results in a learning activity. An instantiation will select the elementary objects needed to construct the unit of learning. Some examples of these elementary objects are now given. It shows the description of the component and its parameters in italics, followed by an example (following each parameter in brackets).It uses the following notation:

? optional * zero or more instances
+ one or more instances - select one of
{} a set of elements<> type of object
@ open parameter
<type>:<name> a type followed by individual instance name

Unit of learning

Unit of learning is a composite learning object that may correspond to a university subject or an update seminar. It is a complex structure that contains meta-data and other components that describe what, why and how the subject can be studied.

-*Learning-unit-name* (data base design)
-*Learning-description* (define relational model, normalization and the SQL language))
+*Output artifacts*(relational tables, SQL programs))
+*learning-content* (background information – reference books)
-*unit-plan options* {Instructivist or constructivist}
-*Learning-evaluation* {formal, informal}

Learning Content *Learning content is collection of learning resources, from on-line learning material, and PowerPoint slides to learning literature. Different types of learning material may also include assessment material like on-line quizzes. Learning content is usually categorized and structured in a way that can be directly accessed from the learning platform (but may also include references like: read the article/book with a given title).*

References:
Text book: Description of organization
+*Output documents*: {relational tables, SQL programs}
+*questionnaires* (assessment of progress)

Learning plans

Learning-plan-name: Design-methodology-1
Learning-objective
+{step 1 – learn about flat files
step 2 – learn about normalization,
step 3 – learn SQL}
}

Learning-Activity *Activity is a formal description of a learning step with a clearly defined goal (as a part of a learning plan). It describes the actions to be performed in a learning step as well as the environment and resources that may be needed to achieve the goal of the activity.*
*{
{*Activity name:* SQL learning instructions – set 1,
Activity objective: Building business models, Learn business plan formulation
Activity type: Group collaboration
Service options: +{meetings, workspace distribution}
Learning content: {case-study-description, business model examples}
+*Output artifact:* Business plan;
**Evaluation criteria*: Evaluation by tutor
**Cognitive tools*: Outlines of earlier studies.
+*Action-objectives*: {discuss alternatives, alternative

analysis, conflict resolution, defining criteria}

Actions *Actions are concrete initiatives to be performed as a part of wider activity, in order to achieve the activity objective. The actions usually refer to system services, like creating a chat room or group assignment communication space, or doing a quiz. Agent matches action objective to that stated in the activity definition.*

Action-name: Carry out questionnaire;
Action-objective: Assess knowledge.
Action-type: +{on-line questionnaire}
+*Service-types*: web
+*Roles:* observer: observes student inputs;

4 The role of agents

Agents are used for two purposes, namely, creating learning of units and then managing the progress of learning.

4.1 Building units of learning

The unit of learning is now built by the individual agents associated with each learning object. The process is briefly illustrated in Figure 4. Here:

- The UOS agent (which is an activity agent) locates the learning content.
- The UOS agent finds a plan from a coordination agent.
- The UOS agent then uses the plan to construct activities by matching step objectives to activity objectives.
- The activity actions are then matched to services and added to the workspace.

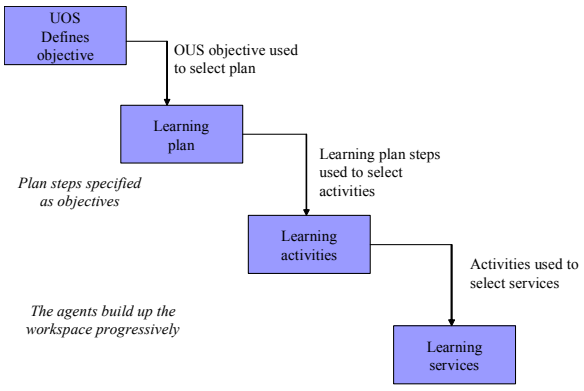

Figure 4 – Constructing learning workspaces

The agents add to the workspace as the process takes place. They also construct the roles and learning content during the construction process. They must ensure consistency between the activities. As an example here are some goals to be met by the building agents.

Subgoal: setup Custom-UOL
If keyword in unit-metadata(learning-objectives) matches keyword in UOL(learning-description) then create custom-UOL from UOL; add UOL(learning-content) to Custom-UOL(learning-content).

Subgoal: setup custom-learning-plan
If sum of learning-plan(activity-objectives) includes all keywords in Custom-UOL(learning-description) and learning-plan(plan-type) matches Custom-UOL(unit-plan-options)
then create custom-learning plan from learning plan.

Subgoal: setup custom-learning-activities
If learning-activity(activity-objective) match custom-learning-plan(activity-objective) then create custom-learning-activity form learning activity, add to learning-activity(learning content) to UOL(learning-content)

Subgoal: seup cutom-learning-actions
If learning-action(action-objective) matches custom-learning-activity(action-objective) then create custom-learning-action form learning-action, add learning-action(roles) to UOL(roles)

The addition here is primarily to workspaces that are constructed by the agents.

4.2 Managing the Learning Process

The agent goals here are now different and center on creating and monitoring learning activities. Each of the objects in the unit of learning can have its own agent. As shown in Figure 5 the usual structure is an agent for a unit of learning follows a plan that includes the completion of a number learning activities. The unit of learning agent delegates work to learning activities by creating a workspace for the learning activity and its agent. The unit of learning agent monitors progress on the learning activity task.

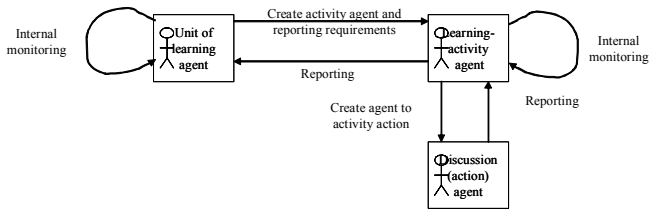

Figure 5 – Architecture of selected agents

At a more detailed level, we use the usual reasoning model of agents where a goal leads to a plan, which is

defined in terms of rules that lead to actions. The plan itself can have lower level goals. The reasoning model is implemented using the three layer architecture [10] chosen from a number of alternative architectures [11]. Agents are used to achieve goals using plans defined by agent users. A plan is composed of event-condition-action rules, each of which specifies the actions to be executed when condition is true. We can predefine the goal, belief, plan, rule, and action for an agent or define them at run time. Following is an example of the goal (including some examples of sub goals), plans, rules, and actions of a simple learning-activity agent for people collaborating on a learning-task.

Goal: Complete learning-activity on time
Plan: for completing learning-activity on time
Sub goal g1: to determine if action is needed to *correctly assign role responsibility*
Sub plan P1: for g1
Rule R1: on timer alter if no role assigned **then** action A1
Action A1: send e-mail message to learner and instructor "you should meet and choose someone as the coordinator for learning activity <learning-activity-name>"
Sub goal g2: to determine if action needed *for work to proceed in a steady manner*
Sub plan P2: for g2
Rule R2: on timer alter if today is later than two days after last update and today is not after the task end date **then** take action A2
Action A2: send the following message to participant in coordinator role "There seems to be no progress on the key document – should it be updated?"
Rule R3: on timer alter if accesses larger than 15 and no updates to key artifact **then** action A3. (probably disagreement).
Action A3: send the following message to participants in team-member "You should begin to make changes to the key document"
Sub goal g3: to determine if action is needed to *improve interaction between team members*
Sub goal g4: to determine if lack of progress *requires some additional learning*

The unit of learning agent would include the rules of actions to take when progress is not as expected. These would include:

Goal: Identify additional learning activity
Plan: for identifying activity
Sub-goal g1: identify need for additional activity
Sib-goal: identify need for additional activity

The actions to be taken would be derived from rules that evaluate reported outcomes from current learning-activities.

5 Mapping models to implementations

We are developing prototype generic agents using our workspace system, LiveNe [12]t. Figure 6 illustrates a typical workspace that supports the creation of learning plan in a unit of learning workspace. The instructor sets up the materials for the unit of learning. The next step is to define the learning activities, including their start and end dates, together with supporting materials relevant to each activity. This in fact becomes the high level process. The activity agent then uses the plan to create workspaces for each work-item and its associated agent.

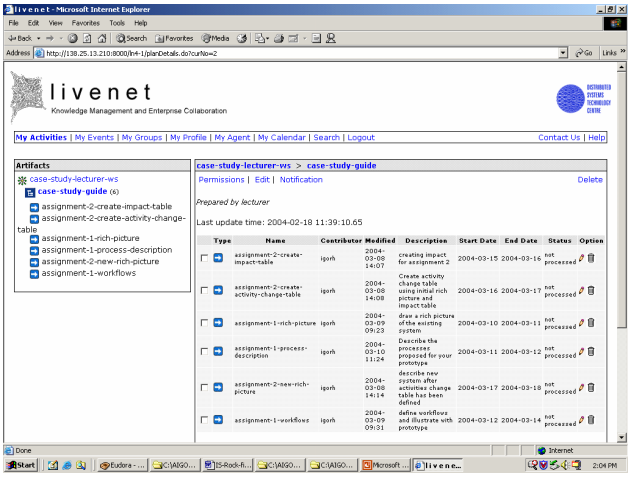

Figure 6– Developing the high level plan in a unit of learning

The workspace shown in Figure 7 is built up by agents as they add activities, learning materials and actions to the unit of learning. Figure 8 illustrates the workspace created for the learning activity. The participants of the workspaces are notified that the task has been created and must meet to decide on the allocation of responsibilities to each team member. The case study agent then uses rules described earlier to monitor task progress.

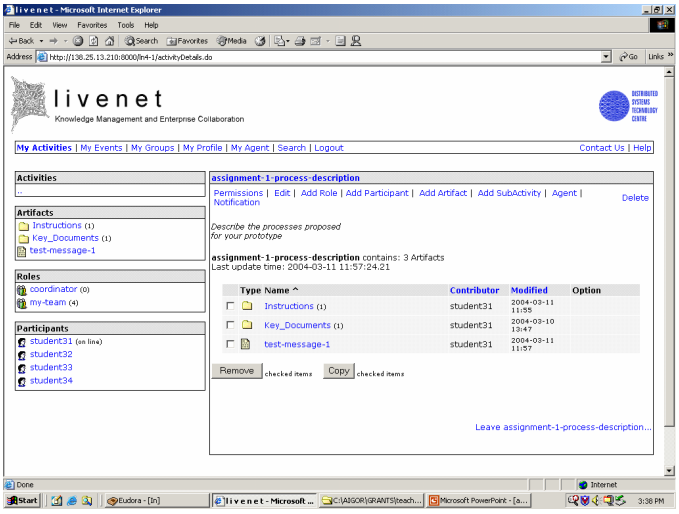

Figure 7 – Case study task workspace created by case study agents

The two workspaces shown in Figures 6 and 7 could, in the generic sense, represent many work situations; for example, the plan could be a software engineering process. The agent for the process then generates workspaces for the individual tasks, allocates people to them, enters the relevant documents and notifies participants to commence their work.

6 Summary

The paper described the importance of active learning systems for personalizing learning experiences. It describes a way to describe learning environments and ways of defining agents that create and manage such environments.

7 Acknowledgement

The research carried out here was supported by the ARC through its Discovery Grant program. Aizhong Lin contributed in the development of the agent system.

8 References

[1] Biggs, J. "Teaching for quality learning at University" Open University Press.

[2] Koper, R. :Modeling units of study from a pedagocical perspective: the pedagogical model behind EML" http:/eml.ou.nl/articles

[3] Hezemans, M. and Ritzen, M. (2002): "Learning Environments: Three types of Learning Environment" Tele-Learning: The Challenge for the Third Millennium", IFIP Congress, Montreal, August 2000, Kluwer Academic Publishers, Boston, pp. 185-192.

[4] Jonassen, D. and Rohrer-Murphy, L. (2002): "Activity Theory as a Framework for Designing Constructivist Learning Environments" ETR&D. Vol. 47, No. 1, pp. 61-59.

[5] Petraglia, J. (1998): "The real world on a short term lease: The (Mis)Application of constructivism to the Design of Educational Technology", ETR&D. Vol. 46, No. 3, pp.53-65.

[6] Dublin Element Set – http://dublincore.org

[7] IEEE Learning Technology Standards Committee – http://ltsc.ieee.org

[8] Fisher, S. (2001): "Course and Exercise Sequencing Using Metadata in Adaptive Hypermedia Learning Systems" ACM Journal of Educational Resources in Computing" Vol. 1, No. 1, Spring 2001

[9] Hawryszkiewycz, I.T. (2002): "Designing Learning Activities from Learning Objects" ASCILITE, Auckland, December, 2002.

[10] Müller J. P. (1996): The Design of Intelligent Agents. Springer Verlag., pp. 7-44

[11] Wooldridge, M.(1999): "Intelligent Agents" in Chapter 1 "Computational Organization Theory" in "Multiagent Systems" Gerhard Weiss (Ed) MIT Press

[12] LiveNet – http://livenet4.uts.edu.au