The current issue and full text archive of this journal is available at
www.emeraldinsight.com/1065-0741.htm

Creating domain independent adaptive e-learning systems using the sharable content object reference model

Jason Watson
School of Information Systems, Queensland University of Technology, Brisbane, Australia
Pervaiz K. Ahmed
University of Wolverhampton, Wolverhampton, UK, and
Glenn Hardaker
University of Huddersfield, Huddersfield, UK

Abstract

Purpose – This research aims to investigate how a generic web-based ITS can be created which will adapt the training content in real time, to the needs of the individual trainee across any domain.

Design/methodology/approach – After examining the various alternatives SCORM was adopted in this project because it provided an infrastructure that makes it possible to deliver personalised learning dynamically using re-usable learning objects.

Findings – The results show that a system which presents a student with content that is supplementary to an authored course should be accompanied by a tool to help the trainee's navigation. For such a tool, key functionality would be: first, to identify learning objects that would take the student towards the ultimate learning goals; second, to suggest a pathway through the authored course structure and additional learning objects to the student; and finally, to present the student with different choices of pathway, such as fastest, most comprehensive and most popular routes.

Originality/value – This investigation has taken another approach of adapting the course by displaying an adapted set of learning objects to the trainee, instead of using a linear course structure.

Keywords E-learning, Training

Paper type Research paper

Introduction: computer based training and online learning

Over the past two decades developments in the area of Information Technology (IT) have led to an enormous increase in the use of Computer Based Training (CBT). Computer performance has, generally, been doubling annually, when taking into account the increased use of extensive hypermedia. The use of hypermedia in CBT has characteristics that provide significant advance for learning, such as integration of different media, provision of a highly interactive environment and use of a non-linear organisation in the form of a network of nodes (Chen and Ford, 1997). The use of hypermedia and multimedia in training applications has made it possible to develop systems which improve learning by making the training interactive and by enhancing the presentation of the curricula with innovative use of graphics, video and sound. Mayer and Moreno (2002) showed how the rate of learning increases when the

Campus-Wide Information Systems
Vol. 24 No. 1, 2007
pp. 45-71

1065-0741
DOI 10.1108/10650740710726482

instruction is performed in this way, by contrasting the performance of students using a combination of listening to narration and viewing corresponding animation with those only listening to narration.

This research investigates how online learning can be improved by using intelligent features in a generic subject domain infrastructure that takes advantage of interoperability specifications. Interoperability standards are exploited to develop a methodology that enables the use of third-party content or content authored by standardised learning content authoring tools. This research was conducted in collaboration with the NTP Group (NTP) between 2001-2004, one of the largest multi-occupational training contractors in the UK, serving 25,000 trainees throughout the UK. NTP has a vision of developing a total intranet solution for the entire enterprise with an intelligent agent that acts as a tutor for the employer in the modern work place. The learning Management System (LMS) is a competency building interface to the services that helps companies to manage its operations, knowledge and training.

E-learning interoperability specifications

Even though CBT has been established for several decades, the e-learning industry is fairly new. The need for e-learning standards has been one of the key issues in the industry for the last decade. As the number of LMS vendors and content developers has increased enormously, the requirement for interoperability has become increasingly important. The choice of an LMS is an important part of an organisation's e-learning strategy and it will affect the organisation for long period of time. Choosing an LMS that restricts the organisation to the use of only the content or authoring tools from the same vendor is a narrow strategy in the long term. An organisation should be able to have a free choice of content from different vendors. This need for interoperability has been the main driving force of the e-learning standardisation process. These standards can also help to establish a base technology infrastructure with *permanency* (Duval, 2001). Rapid development of ICT can cause systems (LMSs, ITS's, authoring tools) to become obsolete, but if the specifications that define the boundaries between infrastructure parts can be relied upon to be robust then long-term adaptive-ness can be ensured.

The current state is that there are no official open standards, but instead various specifications aiming to contribute to official standards. The situation has been confusing for the last few years for those wishing to purchase e-learning solutions, as the industry is filled with buzzwords like "AICC certification", "SCORM compliant", "Supports industry standards" etc., even though customer awareness of the advantages of adopting standards has been increasing rapidly. Different organisations have been involved in developing the interoperability specifications, such as IMS, ARIADE, Aviation Industry CBT Committee, IEEE (Aviation industry CBT (2002), ADL-SCORM (2002), IMS Global Consortium (2002), ISO/IEC JTC1/SC£^ (2002)). Fortunately, in recent years these organisations have been working closely together and interchanging parts of each other's specifications. The main aim has been to eventually incorporate aspects from work done by different organisations into an official ISO standard. After examining the various alternatives SCORM was adopted in this project because it provided an infrastructure that makes it possible to deliver personalised learning dynamically using re-usable learning objects. For the purposes

of this research and development project, the specifications mainly address two specific problems:

(1) how to define the learning object and resource packages so that they can be imported into any LMS; and

(2) how to enable content tracking and run-time communication between the content and the LMS.

Developing an intelligent adaptive learning system

The flexibility and benefits provided by online learning (economies of scale, re-use, ease of updating, wherever whenever access) are well known and this has acted as a strong incentive for institutions to transfer content online as quickly as possible. However, one downside in online learning is that trainees are isolated from the normal human-to-human interactions that are part of the traditional learning process (Skinner, 1958, Kimble, 1961, Houston, 1981, Anderson, 1990). In most LMSs trainees can send queries, questions and exercise results to the tutor, or even "attend" live video-conferencing type sessions, but even so, live contact and the reception of instant synchronous feedback is limited (Alaven and Koedinger, 2001). On the other hand, a classroom tutor is able to do this by answering trainees' questions and otherwise interacting with the class. For the past twenty years, research into Intelligent Tutoring Systems (ITS) has investigated how the absence of the tutor could be compensated for in computer based learning (Alpert *et al.*, 1999, Gehne *et al.*, 2001). Some systems have tried to imitate the actions of the human tutor, but virtual learning environments do not often provide such functionality. Ritter *et al.* (1998)) lists actions that are performed by the tutor during a class:

(1) Provide examples of how to use the core knowledge learnt about the subject.

(2) Provide advice and help for the current learning tasks.

(3) Monitor trainee progress and make corrective actions if trainee loses the track:
   - Notice a hesitation and give hints how to proceed.
   - Ask questions about the subject.
   - Monitor the trainees' preferred learning style and provide the actions in preferred format.
   - Browsing pattern could determine whether the student wants to learn the content in more depth.

(4) Provide more information on the subject for more advanced trainees.

(5) Provide initial help for the questions:
   - By providing access to the information describing the answer.
   - Giving straight answers.

A tutor uses his pedagogical expertise and awareness of students' capabilities to direct the instruction. The challenge is to develop a machine or piece of computer software to imitate these actions. What kind of information should be available for a procedural program to make such complex decisions? How would the machine represent fuzzy entities such as knowledge or trainees' behaviour? To provide advice and help on the current learning tasks requires an ITS to have understanding about the subject area

(Ritter and Koedinger 1995, 1996, Self, 1999). For the ITS to decide what kind of help the student needs, it requires an understanding of the student's knowledge, what problems the student is experiencing and how to present help for the student in a manner that is suitable for this particular individual. The do this, the ITS would have to contain a model of the students' knowledge, expertise in the subject area and pedagogical expertise.

One important aspect of an ITS is the personalisation of the training content; i.e. adapting the content to an individual trainee's needs (Sampson *et al.*, 2002). Commonly, such a system requires some form of user modelling to store information about the user's knowledge, preferences and previous actions. In adaptive educational hypermedia, a student will be given a presentation that is adapted to suit his or her needs, with particular regard given to their knowledge of the subject. Brusilovsky (2001) has reviewed work done on adaptive hypermedia systems and user modelling three times in past decade. He states that: "the way in which conventional hypermedia applications show the same static information to all users has been one of the limitations in the hypermedia". This has been a motivator for research in a number of organisations such as the International Artificial Intelligence in Education (IAIED) society.

Cost-effective solutions using generic domain systems

An adaptive educational hypermedia system can enhance learning efficiency, but for a company to develop such a system other factors and requirements have to be analysed. Most of the ITS's have been developed for a specific problem domain (Mullier *et al.*, 1998), e.g. ELM-ART for delivering instruction on the LISP programming language (Weber and Brusilovsky, 2001). To deliver instruction on other problem domains, a new intelligent tutoring system would have to be developed, requiring an investment of resources of a considerably higher magnitude. To deliver cost-effective, adaptive online learning content the e-learning industry requires a solution where learning content for ITS's could be:

- Authored with off-the-shelf multimedia authoring tools.
- Bought from third-party content vendors.
- Able to address any problem domain or subject.

An ITS which fulfils theses requirements could gain the benefits of increasing learning efficiency in a cost effective way (Blumenthal *et al.*, 1996, Arruarte *et al.*, 1996). However, many intelligent tutoring and adaptive educational hypermedia systems are bespoke applications for a specific problem domain or are authoring tools which can be used to generate ITS's, e.g. TEx-Sys (Stankov *et al.*, 2000). There are some generic adaptive hypermedia systems available (Kobsa, 2001), but these are mainly designed for non-educational applications.

When the ITS uses content in a format defined by the SCORM specification, content can be bought from any vendor producing SCORM compliant content. The main reason that personalised learning technology has not yet reached mainstream status, is the difficulty in re-reusing the research results (Sampson *et al.*, 2002, as discussed in Karagiannidis, 2002). In earlier research he stated:

> Existing standards do not adequately support the definition and interchange of re-usable adaptive and flexible learning methods which are beyond the 'rigid' approach directive, curricular based linear training, as enabled by the envisaged hierarchical structure description in the [IMS] Content Packaging standard (Karagiannidis, 2001).

Thus, SCORM allows for the re-usability of learning objects, but is deficient in the areas required to define reusability, e.g. student and pedagogical models. This research investigates how to exploit the SCORM standards for the use of an intelligent web-based adaptation system, with minimal extensions.

Design options – intelligent course or dynamic course

Once the initial design requirement of using SCORM compatible learning objects within the adaptive system was made, two options for the design were identified:

(1) A system which would pull the learning objects from a central data repository and construct the individual courses for each individual trainee, based on their existing knowledge and learning goals. A linear authored course structure could be used as the basis of the pathway and the system would recommend to trainees different pathways towards the same learning outcomes.

(2) The content author would generate a course structure by defining a network of sharable content objects (SCO) which relate to one another. This Intelligent Course Structure (ICS) would apply for all trainees, but the ITS engine would make decisions as to exactly how the trainee would proceed towards the goals. The ITS engine would, after the execution of one SCO, analyse the users knowledge/goals and learning outcomes to define the next action.

For the first option, it has to be asked if it is possible for a system to generate a consistent course from independent SCOs, based only on the additional meta-data that relates the SCOs to a competency framework? Learning objects are often authored using different graphical templates that are customised to a customer's needs. If the SCOs in the repository have independent looks and navigation, the lack of an overall look for a course could be confusing. SCORM does not, to date, address the need for differentiating the learning content from its graphical user interface.

In the second option, a network of learning objects could be simplified by layering the SCOs (Figure 1). As one proceeds lower down the learning network, the detail

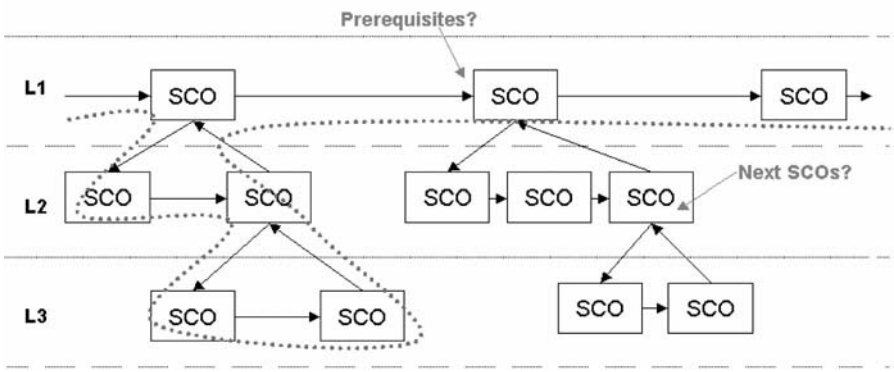

Figure 1.
Layered intelligent course structure (ICS)

within the subject increases. The method that this ITS would utilise to interpret this structure can be best understood with the aid of an example. The system has allocated a specific course for a student. The student starts the course and goes through the first SCO in L1. If the system notices that the trainee has the required skills, the execution flow could continue to the next SCO in L1. However, the system forwards the student towards lower levels and does the same analysis after each SCO. Some SCOs only have one exit point and no analysis is needed. After returning to L1 and advancing to the second SCO, the system analyses that the student has already gained the skills for remaining SCOs in lower levels, or the student does not have to learn such detailed skills, so the ITS forwards the student to the last SCO of the course and exits the course title.

This approach can be related to the normal chaptered approach of authoring a course (Murray, 1999). The five SCOs on the left side can be seen as a first chapter, the SCO in the top L1 layer delivers basic knowledge covering the whole chapter. The structure should be able to find alternative routes for the individuals. This simplified approach only defines the possible exit points for the SCOs, but it could be said that instead of going through the SCO1, the student could take SCOs 2,3 and 4. This would put the pressure on the authoring process as the same subject would have to be authored on different levels, i.e. one SCO for the people already familiar with the subject and one (or several) for people that are unfamiliar. Table I summarises the positives and negatives of the two available options.

Content authoring should not be based solely on the technologies we can use to create and build the application, but should also incorporate the theory of learning and instruction to increase the rate of learning. Many of the first online learning courses were directly transferred from existing lecture material and did not take advantage of new media to create courseware that follows the guidelines of cognitive science and learning theories. Cognitive load theory is one of the most important components that should be taken into consideration in educational multimedia authoring (Mayer and Moreno, 2002). This is based on relating the instruction to the architecture of the brain

	Combining courses from a SCO repository	Authoring an ICS that adapts a course to the individual
Authoring	+ No extra task for authoring. The SCO is defined with learning outcomes and prerequisites and left as it is. The ITS will analyse the SCOs and create a pathway	− The course author needs to create a course network that can be applied for all individuals
Pedagogical requirements	− If the responsibility of presenting the course in a pedagogically sound manner is left to the ITS, then a highly sophisticated expert system is needed (and SCOs have to be tagged in great detail)	+ The course author can use his knowledge to create a logical way of presenting the subject (with adaptation choices)
Fulfilling the training requirements	− The SCOs could be displayed in a non-structured way (random-like order)	+ The learning outcomes are displayed in the context and in right order

Table I.
Comparison of design options

itself. While short term working memory is limited in handling simultaneous multiple elements or facts, long term memory is based on the sophisticated structures of information i.e. schemas (Sweller, 1999), developed by Mandler (1984) and Rumelhart (1980). Memory takes the form of schema which provide a mental framework for understanding and remembering information. When people are shown a picture they remember different things about it according to the different schemas they have developed. Therefore schemas can be seen as influential elements in cultural differences in cognition (Quinn and Holland, 1987). "Research on novice versus expert performance suggests that the nature of expertise is largely due to the possession of schemas that guide perception and problem-solving." Cognitive load theory recognises that in instructional design the emphasis should first be on developing the appropriate schemas, and then proceeding to more advanced learning in the subject. In this way the student would set up their brain architecture to digest the information prior to proceeding to more detailed information. If these schemas are not developed, the student will process the presented information in their short-term memory, but will not retain it in their long-term memory. Sweller recommends four principles for designing instructional material:

(1) The development of schemas should be supported by presenting goal-free problems or examples with solutions, to save the resources of the working memory.

(2) The information sources should be integrated physically to present the student with a good overall view of the material.

(3) Redundancy of the information should be reduced to reduce the load of information.

(4) Information should be presented in integrated visual and auditory format, but not if it is gratuitous.

Sweller's guidelines are good instructions, not just for authoring, but for overall system design as well. The first argument mainly relates to the authoring process and suggests that authors should use practical everyday examples whenever possible. Prior to the exercises and tests the trainee should be presented with a case study or similar information giving him/her an overview of the subject without having to engage with little details that are not necessary for understanding the big picture. In a system that dynamically schedules small learning objects and recommends related learning objects to the trainee, these example SCOs should be authored as separate SCOs. A system could then recommend a course structure with the core SCOs in the subject areas, and then possibly recommend additional SCOs with related case studies and examples. Sweller, secondly, recommends that the information sources should be physically tied into each other to give the student an overall view of the subject. In SCO authoring this could mean the use of mind-map type hyperlinked pictures representing the subject area as a whole. This argument can also be analysed in the larger context of illustrating the whole course and related information sources. It could be assumed that a graphical tool that visualises the whole course content and additional information sources to the trainee would be beneficial towards better learning. The trainee should have a clear overall view of the whole subject area he/she will be going through with all additional information sources that can be accessed. We touch upon this area later in

discussing the development of a *User Interface* or *Learning Navigator* interface. Sweller, also, notes repetition as confusing for the trainee as it can overload the short-term memory. Again this can be used directly as advice in SCO authoring, but it also applies to system design. If the trainee has learned new skills in the early parts of the course, the system should not recommend more SCOs consisting of similar skills.

The next task was to analyse what could actually be adapted and then decide what adaptation was to be incorporated into the development.

What and how to adapt?

Adaptive Educational Hypermedia System (AEHMS) is a system which adapts the learning information to the needs of an individual. The adaptation can be at any level, e.g. the user may be presented with a list of links based on his/her past actions. The past actions are monitored by the system and stored in a student model. The student model defines the student's knowledge or interest in different subject areas. The whole process naturally depends on the system and the requirements of exactly what has to be adapted and on what principals the adaptations are based on. An AEHMS may have several variables upon which the system bases its adaptation decisions. Brusilovsky's (2001) review identified the most used variables as:

- user's goals/tasks;
- knowledge;
- background;
- preferences;
- user's interests; and
- individual traits (e.g. personality factors, cognitive factors, learning styles).

Brusilovsky (2001) defines a taxonomy for categorising the type of content adaptation. He defines two main level categories as *content level adaptation* (or adaptive content) and *link level adaptation* (adaptive navigation support) to distinguish two different areas of adaptive behaviour. The first one includes systems that actually modify the content based on the ITS decisions, e.g. by changing the content of text paragraphs or changing the type of the content from text based to more visual. Link level adaptation systems use hypermedia links to display different learning content to trainees. Adaptation is accomplished by hiding, disabling, sorting or dimming the links based on their relevancy to the individual user. As this research respects the SCO boundary to take advantage of the existing SCORM compatible authoring systems, it therefore does not reflect either category. To allow for use of existing learning objects in an ITS, the smallest adaptation level would have to follow the granularity of the learning objects. The system would more probably adapt the whole course structure to the needs of the individual than adapt the content inside learning objects. Thus, a third main level category would be needed to describe systems that alter the path through existing content in a learning object repository. Such a category could be named as "Adaptive content scheduling systems".

Based on the assumption that the SCO is the smallest size of adaptation, we will now define a few essential items that will be used in the following discussion. In traditional systems an author creates learning objects and combines them into a linear course structure. This authored path is created using the pedagogical and subject

expertise of the author and therefore contains the learning objects that the author considers most crucial. We call these learning objects "Stage SCOs". The system will suggest additional learning objects to the trainee which relate to the Stage SCOs. These SCOs are external from the original course structure and we call them "Proximate SCOs". When references are made to a "Pathway", this refers to a route through the course Stage SCOs and Proximate SCOs.

The student model

The student model (or user model) can be seen as the information storage that defines a student against the variables upon which adaptation decisions are based. Following Brusilovsky's (Brusilovsky, 1996; 2001) most common attributes for decision making, the student model would store such information as: where the student is (knowledge), where the student wants to go (goals, tasks and interests) and how the student prefers to arrive there (background, preferences and individual traits). It could be said that all adaptive hypermedia systems have a student or user model in one form or another. Educational applications can have very complex models for the user's current knowledge and learning styles (Felder, 1993, Felder and Silverman, 1988). A student model can also be simplified, for example stating whether a user prefers to see the information in highly audio-visual format or in plain text.

The individualisation of the course to a student's needs could be primarily based on user preferences. If the user prefers to see the content in a text, rather than in more visual format, the learning content repository would have to contain two SCOs with identical information, but different levels of graphical visualisation, or one SCO would have to deliver both formats. Even though benefit to learning efficiency could be gained when the student is allowed to take a course in the preferred format, this would increase the development effort. Further analyses would be required to identify whether the benefits gained are be cost effective. This specific analysis is not included within the scope of this paper.

The project challenge was to develop a system where SCOs could be dynamically fetched from the learning object repository to provide an individualised pathway towards the same learning outcomes as an authored course. It assumes that not everyone has to take the same SCOs to achieve the same outcomes and some students need to take additional SCOs to gain a comprehensive understanding of subjects in the learning content. To achieve this end, using only learner preferences is clearly insufficient and therefore other attributes are needed. Requirements regarding the student working towards learning outcomes clearly state that the learning goals (or tasks, interest) have to be included in the student model. By doing this, SCOs can be suggested to the trainee based on how the learning outcomes of one SCO relate to the learning outcomes of the whole course. Learning outcomes of the course can be seen as current goals of the trainee even though the student may have other goals as well. It is also required that the amount of existing knowledge in the subjects of the SCOs has to be measured so the current knowledge of the student in some subjects has to be incorporated into the student model.

Learner preferences and learning styles, though important aspects did not feature in the initial remit of this project. Thus they are not discussed in this paper, even though both are important attributes parts of the whole context. This investigation started by evaluating the most essential attributes and trying to build-up a prototype from this

base. This research takes the student's knowledge and goals to develop a student model which fits into the SCORM specification. If the learning objects are recommended to the trainee based on the trainee goals, e.g. course outcomes, then there has to be a way to map the SCOs to the learning goals.

A student model has to contain a representation of the level of student knowledge for each SCO in the repository. To store the level of knowledge directly against the SCOs is inefficient, as the SCO can contain elements from different areas. At this point in the project we defined a "Skill" attribute within the student model. The structure of a skill allows mapping of the learning outcomes to the goals, as a skill can be a learning outcome and also a goal. Learning objects can contain many skills each with a different weight and a course can have many weighted skills as its learning outcomes. The student model identifies the level of a student's knowledge in each of the skills. If it can be assumed that a student masters a skill, the skill level is set to 100 per cent and if the student does not have any knowledge in the subject area that the skill represents, the skill level is set to zero. The skills also relate to each other as they do in real life. An "Information Technologies" skill can contain other skills within it, such as "Database" *skill* and "Programming Languages" *skill*. Figure 2 illustrates the structure of the skills hierarchy in the proposed student model, using the example of hierarchy under the "Microsoft Products" skill.

To summarise the definitions and facts of our student model:

- a SCO is associated with one or many skills (with different weights);
- a course is associated with one or many skills (with different weights);
- course skills are considered as trainees' current goals;
- the student has a knowledge level for each defined skill;
- a skill can be a category that contains other skills under it[1]; and
- skill can belong to an upper category (that is also a skill on its own).

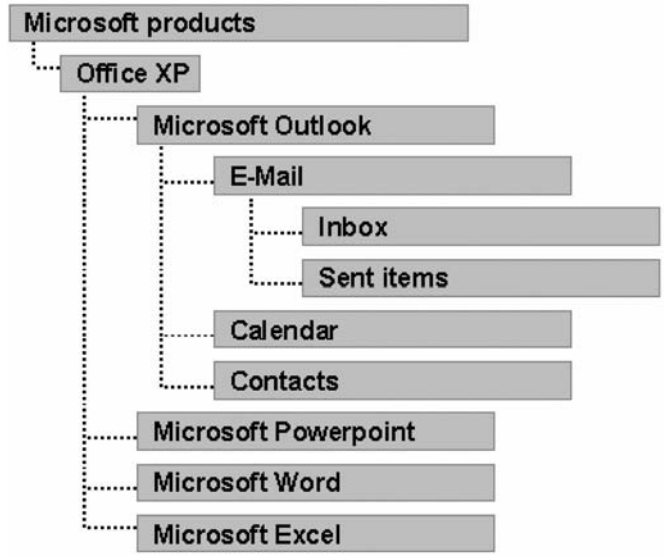

Figure 2.
Example of the skills hierarchy

Routing engine

The routing Engine is a module in our infrastructure that makes the decisions about the learning adaptation. The decisions made by the routing engine loosely follow the framework used by Karagiannidis *et al.* (2001b)). They define a layered structure for the evaluation of adaptive and personalised learning services, which use two distinct high level processes: Interaction Assessment and Adaptation Decision Making. The Interaction Assessment phase makes high-level conclusions from the learner's interactions during their learning. These conclusions are analysed by the Adaptation Decision Making component, which personalises the content for the trainee. This research re-defines this infrastructure for the use of dynamically scheduling SCORM compatible learning objects. The Interaction Assessment is defined as SCO Performance Assessment and the Adaptation Decision Making element as a Pathway Generator. These two modules, together with the student model, form the Routing Engine.

SCO performance assessment

The SCO Performance Assessment module (SPA) is responsible for collecting and analysing the information that SCOs send to the LMS, and making decisions on how this represents student's knowledge in the skills associated with the SCOs. The SCORM data model defines what kind of information can be stored in the LMS from a SCO, such as cmi.core.score.raw and cmi.interactions.3.result. From the data model, the interaction data and the core data can be used to analyse student performance. Interaction is a transparent element that is defined by the SCO author. One SCO can contain any number of interactions, which all have certain attributes (defined in Table II). Interactions are always tied into a SCO, so two interactions in different SCOs with the same interaction ID are considered as totally separate entities. All data stored about the interactions is stored against the student who is currently working on the SCO.

Element	Function
_children	Return string containing elements that are supported by LMS
_count	Returns number of records in the cmi.interactions list for current SCO/trainee
n.id	Unique SCO-specific identifier for an interaction
n.objectives._count	Returns number of elements in n.objectives list
n.objectives.n.id	Developer created identifier for an objective
n.time	Identifies when interaction was created
n.type	Identifies the type of the student response, e.g. "true-false", "choice" or "fill-in"
n.correct_responses._count	Returns number of elements in n.correct_responses list
n.correct_responses.n.pattern	Description of possible student responses to the interaction
n.weighting	Relative importance of specific interaction
n.student_response	Actual student response to the interaction
n.result	Result of interaction, set by the SCO
n.latency	Time from presenting the stimulus to the completion of the measurable response

Table II.
SCORM interaction elements

For example, if a SCO contains two interactions at a certain point of its execution, it can call the following API commands to set the interaction results into the LMS:

(1) n = LMSGetValue("cmi.interactions._count")
(2) LMSSetValue("cmi.interaction.n.id","INT_001")
(3) LMSSetValue("cmi.interactions.n.student_response","No")
(4) LMSSetValue("cmi.interactions.n.result","correct")
(5) $n = n + 1$
(6) LMSSetValue("cmi.interaction.n.id","INT_002")
(7) LMSSetValue("cmi.interactions.n.student_response","No")
(8) LMSSetValue("cmi.interactions.n.result","wrong").

The first command in this example gets the number of interactions that are stored in the LMS against this student/SCO. In this scenario, the new interaction results are stored every time the student executes this particular part of the content. The first time the student accesses the SCO, "n" will be assigned as zero, as no interactions are found on the LMS. Commands 2-4 are all setting attributes of the same interaction, with ID being INT_001, student response equal to "No" and a result of the interaction as "Wrong". Line 5 increases the interaction counter to "1", thus commands 6-8 store their data as a new interaction. If the user comes back into the SCO and the same commands are executed again, the LMSGetValue returns 2 as a count of the existing interactions and the new interaction results are set into slots "2" and "3" (using a zero based storage model). The resulting data in the LMS would look like the one shown in Table III, assuming that the second time the user answered correctly, yes, to the second interaction.

If the SCO would prefer to overwrite the answers for each interaction every time the student comes back to the SCO, fixed position numbers could be used inside the SCO, instead of querying for the cmi.interaction._count. The interaction information is useful for the SPA as it can analyse how many of the interactions have been performed correctly. This mechanism provides the SCO author with a powerful means of effecting the evaluation of the SPA. The effect of this on pedagogical quality will be evaluated later.

In addition to interactions, the SCORM core data provides a useful evaluation element cmi.core.lesson_status. The SCO has a responsibility for evaluating itself and defining whether the student has passed or failed the SCO. Depending on whether the SCO that is being considered is "passed" (cmi.core.credit) or not the lesson_status can be set to passed/failed or complete/incomplete respectively. Again it can be observed that greater responsibility is left to the author to ensure that correct evaluation patterns are used inside the SCO. Also, at this point it has to be noted that not all the interactions in the SCO can set lesson_status to "complete" or "passed" when the user has navigated through all of the material in the SCO.

Table III.
Results of the cmi.interactions example

Nr	Id	Student response	Result	Others ...
0	INT_001	No	Correct	< EMPTY >
1	INT_002	No	Wrong	< EMPTY >
2	INT_001	No	Correct	< EMPTY >
3	INT_002	Yes	Correct	< EMPTY >

Figure 3 illustrates four steps that the SPA makes in the process of deciding the user's performance on a SCO. The process is initialised when the SCO finishes and calls the API's LMSFinish command. The four steps are elaborated further below.

Count interactions result: R_{INT}

SPA goes through each new interaction that is assigned to a SCO, which forms part of the criteria required to finish, and counts up the total interactions result. Only the interactions that have been added during the execution of the SCO will be counted. Interactions in which cmi.interactions.n.weight value equals 0 are not counted in the result and if no weight is set, a default of 50 is used. The cmi.interactions.n.result values are converted as shown in Table IV.

Finally, the relative result of interactions is counted by dividing the sum of the single weighted interaction results by the maximum possible result. The following formula is used:

$$R_{INT} = \frac{R_{Total}}{R_{Max}} = \frac{\sum Result_{weighted}}{\sum Weight}$$

Count lesson_status *result:*R_{LT}

The data model element cmi.core.lesson_status is converted into a value as shown in Table V.

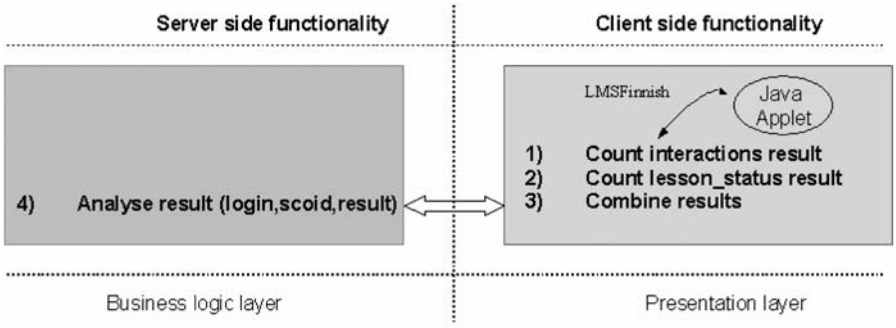

Figure 3.
SCO performance assessment (SPA)

Value	Converted to:
"correct"	75%
"wrong"	15%
"neutral"	50%
"unanticipated"	0%

Table IV.
Converting interaction result to value

Value	Converted to:
"completed" or "passed"	75%
"incomplete" or "failed"	15%

Table V.
Converting lesson_status to a result value

Combine results to final result: R_{COMB}
Counting an average value of interactions and lesson status combines the intermediate results to the final result:

$$R_{COMB} = \frac{R_{INT} + R_{LT}}{2}$$

However, if there is no interactions result, then only the lesson status result is used i.e. do not count the average with zero:

$$R_{COMB} = R_{LT}$$

The final combined SCO result is passed into the server to be analysed.

Analyse results against the skills
The server side functionality of the SPA gets a result (0-100) for one SCO and analyses it against all skills associated with the SCO. The analysis uses the following formula to calculate the new knowledge K:

$$K_t = R_t(1 - K_{t-1}) + K_{n-1}$$

where:

- K_t Knowledge level at time t
- R_t Result at time t
- t increased by 1 when R arrives to SPA

Figure 4 presents how the knowledge (triangles line) increases as a student goes through the learning objects and receives a series of results towards a particular skill (square line).

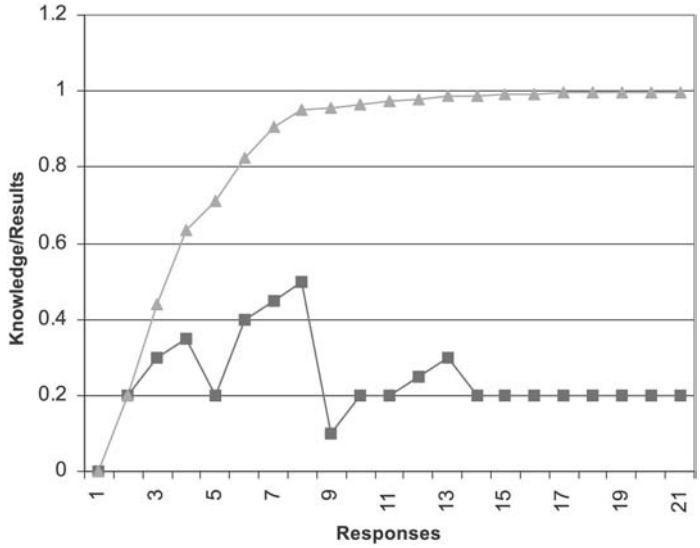

Figure 4.
Accumulative knowledge

Pathway generator

How would the student select the correct learning objects to achieve the learning objectives, when assuming that he/she has access to the repository filled with various learning objects? If the student is allowed to pick learning objects, the pathway could become illogical and confusing. On the other hand, if the student followed a static linear pathway through the learning objects, the outcome would not necessarily match his/her individual needs. Dictating the path cannot be seen as beneficial towards learning efficiency. However, allowing the student to have the freedom to select learning objects outside the original path has proven to be beneficial (Fischer and Scharff, 1998). Investigations have been made into how to help students navigate effectively through existing web-based learning resources and led to the development of a Pathway Planning Assistant (Suzuki *et al.*, 2001). This system provides a student with the tools to plan the pathway prior to actually starting the course. The results of the study were encouraging and proved that the use of the Planning Assistant enabled a student to efficiently direct their attention to pages which fulfil their specific learning goals.These results show that a system which presents a student with content that is supplementary to an authored course should be accompanied by a tool to help the trainee's navigation. For such a tool, key functionality would be:

- to identify learning objects that would take the student towards the ultimate learning goals;
- to suggest a pathway through the authored course structure and additional learning objects to the student; and
- to present the student with different choices of pathway, such as fastest, most comprehensive and most popular routes.

These requirements would also generate a system which adheres to the constructivist learning theory (REF?), as the student would be part of an environment where he/she would have to play an active part in selecting the learning objects. The system recommends a pathway to the student, but not actually dictate the learning process.

When calculating the relevancy of the SCOs, several variables have to be noted. The Stage SCO is more relevant, as the student has less knowledge of the skills associated with it. This represents the basic part of the formula where the amount of skill is divided by the student's knowledge of the skill. The basic element is the sum of all skills associated with the SCO. Factor f_a is used to weight the importance of these elements:

$$R_a = \frac{\sum^{\text{all skills}} \left(f_a (A/K) \right)}{n_{\text{all skills}}}$$

In addition, the Stage SCOs gain more relevancy when they are linked to the goals i.e. to the learning outcomes of the course. In this case, the sum of the elements with the linked skill associations are added to the basic elements. Factor f_{lg} is used to adjust the importance of the linked skill elements. The formula for calculating the relevancy of the Stage SCO is:

$$R_{ss} = \frac{\sum^{\text{all skills}} \left(f_a (A_{ss}/K) \right)}{n_{\text{all skills}}} + \frac{\sum^{\text{linked to goals}} f_{\text{lg}} (A_g/K)}{n_{\text{linked to goals}}}$$

When calculating the formula for ascertaining the "Proximate SCO" relevancy, the previous elements are used in addition to the linked elements in the Stage SCOs. The factor f_{lss} defines the power of the elements linked to the Stage SCO. The final formula is as follows:

$$R_{ps} = \frac{\sum^{\text{all skills}} f_a(A_{ps}/K)}{n_{\text{all skills}}} + \frac{\sum^{\text{linked to goals}} f_{\text{lg}}(A_g/K)}{n_{\text{linked to goals}}} + \frac{\sum^{\text{linked to stages cos}} f_{lss}(A_{ss}/K)}{n_{\text{linked to stages cos}}}$$

User interface

Based on the definitions of the SPA and the PG, the process of adapting the content for the trainee is accomplished by presenting a student with Stage SCOs and proposed Proximate SCOs along with the suggested pathway that takes the student through the course structure. We define the third part of the infrastructure as a "Learning Navigator", which displays the whole course structure with the Proximate SCOs and the proposed pathway to the user. The collaborating company NTPIS initiated a parallel project with the Turku Polytechnic (Finland) to develop a Graphical User Interface (GUI) to exploit the benefits of visualisation of the adapted course structure. Details of this are described in Narinen's (2002). This project took advantage of the system developed as part of this investigation and uses the output of the Pathway Generator. The information that needs to be conveyed by the Learning Navigator is defined below (Kuisma and Watson, 2002):

(1) Stage SCOs.
(2) Proximate SCOs.
(3) Course outcomes (skills).
(4) Proximate SCO relations to Stage SCO.
(5) Relevancy of the SCO (which depends on):
    - trainee level of the SCOs (Skills associated with the SCOs);
    - SCO relevancy to the course outcomes; and
    - SCO relevancy for the Stage SCO.
(6) Different paths (with the order of the SCOs).
(7) Current course position.

The Learning Navigator displays an authored path to the student: the Stage SCO with the group of Proximate SCOs for each Stage SCO. Each SCO is displayed with relevancy, which is calculated from the student's knowledge level of the skills associated with the SCO compared to the skills that are allocated as course outcomes and Stage SCO skills. The learning Navigator also displays the suggested path through the whole course structure.The example demonstrated in Figure 5 is part of a course on "Protecting individuals form abusive behaviour" consisting of three Stage SCOs. Each Stage SCO has been grouped together with additional Proximate SCOs. If the path viewed was the default path (original path authored by the training manager) it would be a straight line. However, the path deviates from this and shows a path that has been chosen by the Routing Engine. The size of the graphic representing the SCO identifies

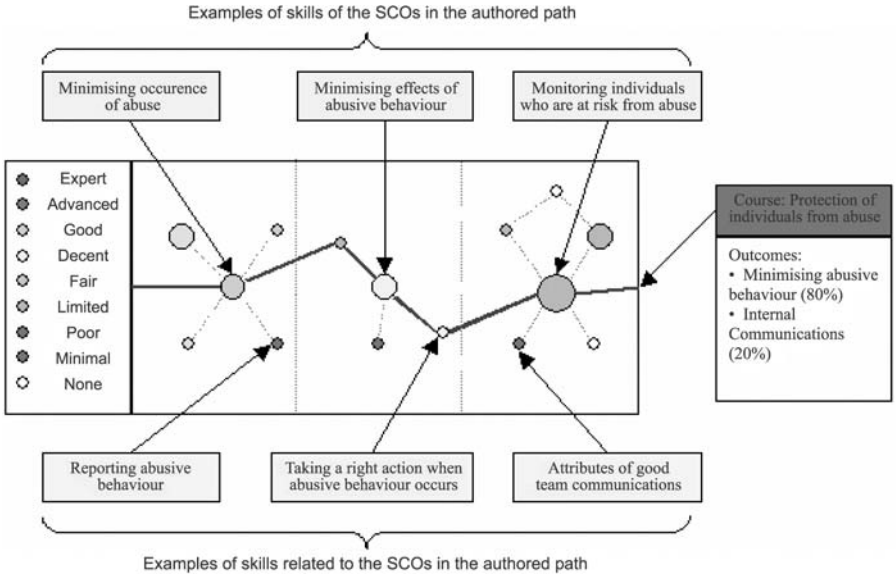

Figure 5.
Learning navigator design prototype

the overall relevancy of the SCO. The colour of the SCO identifies the users' knowledge of the skills associated with the SCO. Colour grading on the left frame describes the meaning of the colour codes, e.g. the trainee has a *good* knowledge of the skills associated with the first stage SCO: "Minimising occurrence of abuse". The Learning Navigator is a trainee's tool to get all the necessary information about the course structure and its different entities. This information can be divided into two parts, visual data and text-based data. The text-based data can be displayed in "roll-over" popup windows and contains detailed information about the SCOs' Skills, and Learning Outcomes (e.g. name, description etc.).

Figure 6 shows the first working prototype of the Learning Navigator (Narinen, 2002).

Requirements for the authoring process

Any of the SCORM compliant authoring tools can be used to create new content or purchase content from a third party vendor. The authoring process of content cannot be restricted by the developed infrastructure. It is assumed that the Routing Engine can freely access the learning object repository. The logical elements of the implementation infrastructure are not discussed, but at this point it is essential to understand that the Routing Engine is an integral part of the LMS, which again is closely connected to the repository.

When an author is developing a course, he/she uses any authoring tool to develop the actual SCO, and then uses "SCO Management Interface" to import it into the LMS. As stated, SCOs are assigned with the skills, and also given a certain weight. The author uses the SCO management interface to add new skills to the SCO with a certain weight. Skills are predefined in the system and the administrator and content authors can add new skills to the system and change their position in the hierarchy. The following example illustrates the whole process:

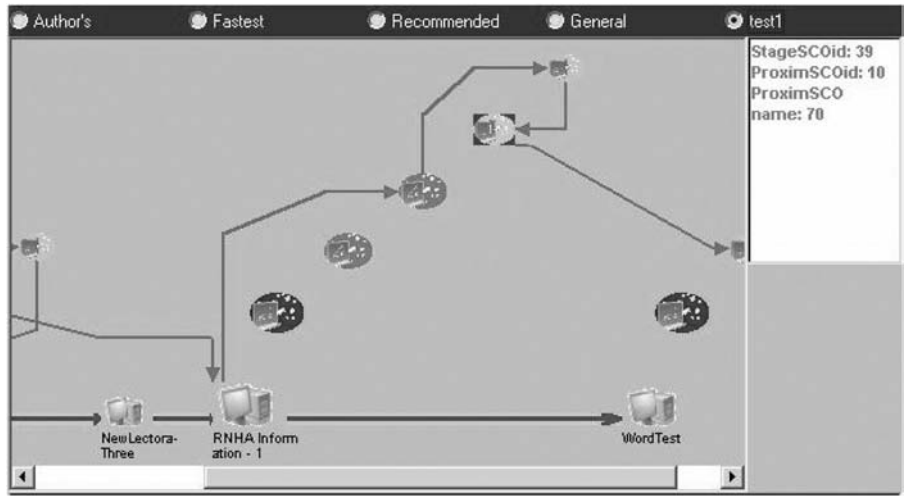

Figure 6.
Learning navigator working prototype

Source: Narinen (2002)

- Add SCO to repository (SCO management interface).
- Add new skill to the system (SCO management interface).
- Assign skills to the SCO (SCO management interface).
- Input the weight for each skill (SCO management interface).
- (Repeat steps 1-5 for each new SCO).
- Create a new course (Course management interface).
- Add SCOs for the course (Course management interface).
- Add skills as the course outcomes (Course management interface).

As mentioned earlier, the correct assignment of the skills is essential to the correct operation of the routing engine. If the author defines a SCO with a few skills with 100 per cent weighting and a student passes the SCO, the SPA will assume that the trainee has fully mastered those skills and no more learning objects will be recommended that relate only to those skills. Problems arise when the author makes it easy to pass the SCO and no interactions are used to analyse how the student has performed. These issues were addressed with the development of simple *Quality Assurance* option that disables the SPA operation for the SCOs that has not been assessed by a subject expert or pedagogically experienced tutor.

For a tutor to evaluate the instructional aspects of the SCO and its skill association, the quantity of the skills the SCO teaches must be measured. For a skill which is entirely encompassed within a single SCO and is defined by a small quantity of knowledge, it may be correct to assign a weight of 100 per cent. If a skill is more general, for example "Microsoft Office XP", it probably is not correct to have a SCO which teaches every single feature of the whole subject area; this may consist of several complex applications and add an enormous amount of functionality. It could be

assumed that the "Microsoft Office XP" skill could be split down to smaller and smaller skills. One SCO could for example contain:

- 100 per cent on "Different views of the MS Outlook Inbox";
- 10 per cent on "Microsoft Outlook E-Mail";
- 5 per cent on "Microsoft Outlook"; and
- 1 per cent on "Microsoft Office 2002".

Implementation of the infrastructure

We now discuss the different choices taken on how to implement the system and outlines the software development technologies considered.

ITS's require a high level of information exhange between the courseware and itself. The courseware and the Learning Management System (LMS) will follow the SCORM specification, which defines a standard way for LMSs to launch the courseware and track a student's performance through the course. This applies certain limitations on the implementation options, as all communication made between the LMS and courseware must be made through the SCORM API, which has to be exposed through the Document Object Model (DOM). Various programming languages, such as JavaScript, Active Server Pages and Java, can be used to create architecture where the API object is exposed through DOM. Alpert *et al.* (1999) analyses different implementation options for intelligent educational applications.

The overall architecture of the system is illustrated in Figure 7. The NTP Wisdom[2] system uses Microsoft Active Server Pages (ASP) for the basic LMS features and Microsoft SQL Server as a database "back-end". Internet Information Server (IIS) hosts the ASP pages and Tomcat is used to host the Java servlet. This server software, together with the server-application makes up the NTP-Wisdom LMS solution, along with the intelligent features which provide the content adaptation. The system is accessible from NTP's LAN or from the Internet through the firewall.

Figure 8 further details the organisation inside the NTP Wisdom architecture. The data layer (database, stored procedures etc.) manages the data in the database. The business layer is divided into two parts, the basic LMS hosted by the IIS and the intelligent functionality and SCORM operations in Servlet hosted by the Tomcat. The client side functionality is implemented in the Java Applet, which is downloaded into the browser when the user accesses the system for the first time. As a comparison to a two-tier system, where all functionality would be implemented in the applet, the server side business functions can be changed without the need to repeat the download. ASP generates the HTML page delivered into the client browser and has a link to the Applet

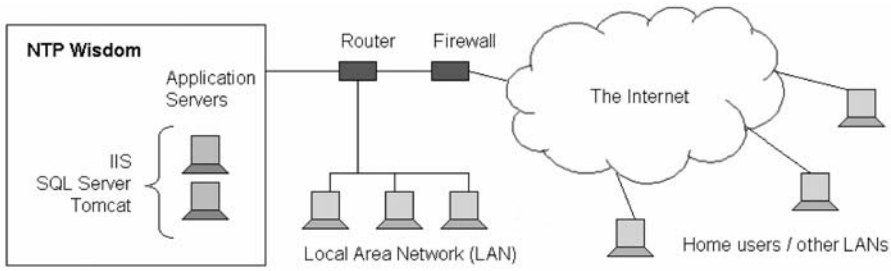

Figure 7.
Overall system architecture

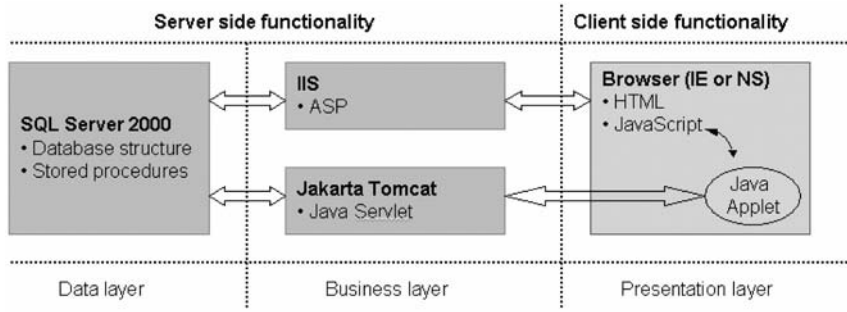

Figure 8.
Three-tier architecture

object, which again is named "*API*" and exposed in the DOM to meet the SCORM RTE requirements.

Figure 9 represents the LMS side and contains four logical elements: Login Validation, My Learning, Administration and Collaboration. These elements can be seen as the most essential parts of the LMS, but as mentioned, the LMS is highly extensible and new elements can be added as needed. The Login Validation analyses the user input for login and password, approves the login process and thereafter builds the interface for the user depending on the user and group configurations. My Learning presents the user with allocated courses, generates reports on the user's performance and provides all the user interface functionality required by the trainee. Administration manages the users, interfaces, learning objects and courses. Collaboration provides trainees with the tools, such as threaded discussions and chat, required to collaborate with other trainees. Content, which is displayed to the users in a separate Learning Window, contains references to SCORM commands that call the LMS API commands. This architecture provides all basic LMS functionality for delivering, managing and monitoring online training but does not include the parts required for adaptive training.

The "intelligence" of the system i.e. the primary focus of this development research project is implemented in the routing engine. The routing engine makes the decisions on learning adaptation using a combination of Interaction Assessment and Adaptation Decision Making as shown in Figure 10. High level events (such as a score obtained by

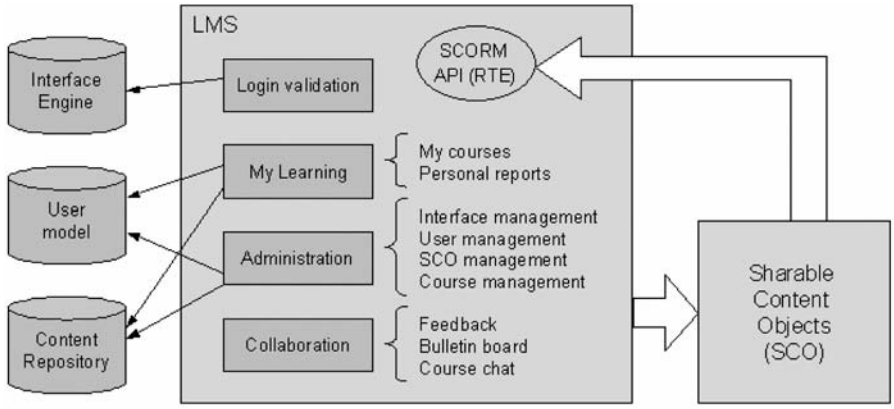

Figure 9.
LMS diagram

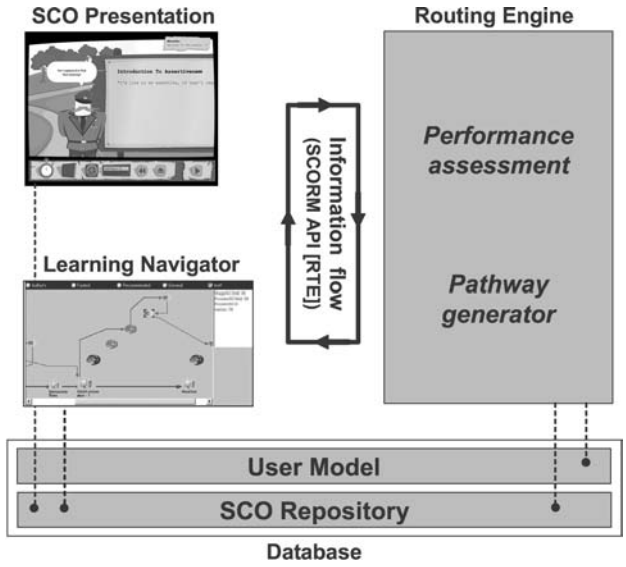

Figure 10.
Overview of path adaptation by the routing engine

a student in a SCO) are analysed by the Interaction Assessment component that makes assertions used by the Pathway Generator (Decision Making Component). One example of such an assertion is a statement on the learner's competency at a specific skill (accumulated by prior experience or good test scores on SCOs related to a specific skill). Assertions are stored in the learner's User Model and accessed by the Pathway Generator. The Pathway Generator compares the user model with the learner's learning objectives and assesses the relevancy of the contents of the SCO repository. The Pathway Generator uses the information on SCO relevancy to create a recommended path for the learner to follow. This path is passed to the Learning Navigator: the interface used by the learner to visualise their course and to navigate their way through the contents of the SCO Repository. This cycle ensures that actions taken by learners whilst performing activities within a particular SCO can influence the future path presented to them. In this implementation we limited the information passed between each component to SCORM data items in order to ensure compliance with the standard. However, the disadvantage of this approach is that low-level events (such as mouse clicks, hover over etc) will not be passed to the Routing Engine and will not be used to by the Performance Assessment component.

Conclusions

Adaptive educational hypermedia systems traditionally make changes to the content that is displayed to the user. For example, the virtual page that ELM-ART displays to one trainee might be different to one displayed to another trainee. The adaptation is accomplished by modifying the page to contain additional information that is personalised to the trainee. This investigation has taken another approach of adapting the course by displaying an adapted set of learning objects to the trainee, instead of using a linear course structure. By doing this, we have retained the benefits of using standards (SCORM) compliant authoring tools instead of bespoke authoring tools or

templates. This is an important advantage, in terms of reduced content development time and the option of using existing content provides direct cost-savings.

By strictly following the SCORM specification, some approximations has to be made for the pedagogical effectiveness of the system. A SCO is an independent learning object; a black box that does not relate to the context it is used in. To follow the specification, the boundaries of this black box have to be respected, which means that no additional data can be assigned to the learning object. Additional information on the SCO is added outside the authoring process i.e. inside the scope of the LMS. A SCO is associated with the skills information to provide the Pathway Generator with a means to analyse the SCOs, which are then proposed to the user. The SCO Performance Assessment monitors the lesson status and the responses to any interactions to ascertain a result for the user. As there is no appropriate means to map interactions to skills, an approximation has to be made to average the result for all the skills associated with the SCO. Further developments to improve this are being made but cannot be addressed here.

The use of learning objects also restricts the level of adaptation, or more probably, the required level of adaptation granularity restricts the authoring of the learning objects. The size of the learning objects defines the granularity of the adaptation. If learning objects are authored as big, one hour long lessons, the Pathway Generator has minimal option to affect the course content. Its responsibility changes from adapting educational content, to the generation of "course" pathways which the user is recommended to take, which it was never been designed to do. The nature of the Pathway Generator is to provide the student with a display of the course content and propose additional information that will take the trainee towards his/her learning goals. The student can choose to pick those additional pieces or ignore them. In NTP, this has led to the development of smaller learning objects instead of "whole-course-in-one-object" SCOs. The answer to the dilemma regarding the learning object size does really relate to its original meaning; being the smallest logical unit of learning. It is supposed to contain one logical item, lesson, unit or subject, as long as it is an independent entity that can be used regardless of its context. If SCOs are developed in this manner in small, logical, re-usable pieces, they will be at their most effective in the Pathway Generator.

One concern regarding the developed content is the way independent learning objects are mixed and matched to produce a logical course. If the content is authored with different templates and possibly purchased from different vendors, the outcome may look confusing. To deliver a consistent graphical representation, the best results would be achieved if SCOs were authored using similar templates. Even though the graphics and colour schemes inside SCOs would change, it would be beneficial to have a consistent positioning for navigation controls. Naturally this cannot be avoided if third-party content is mixed in the system. The solution might be to categorise the content and only allow content authored using the same set of templates to be mixed.

The choice of using SCORM as an interoperability specification proved to be advantageous during the duration of the research project. New SCORM compliant authoring tools have been identified constantly throughout and most of the popular LMSs have adopted SCORM. One of the research goals was to investigate the possibilities of delivering cost effective personalised learning to meet industry requirements. To achieve this, it was essential to define an architecture which relies on technologies that have been widely adopted by the e-Learning industry. SCORM has proven to be such a specification and it will provide a good platform for future work.

Notes

1. The knowledge in child skill does not accumulate towards the parent skill.
2. NTP Wisdom is the proposed marketing name for the developed system that includes the LMS and Intelligent Tutoring functionality.

References

Alaven, V. and Koedinger, K.R. (2001), "Investigation into help seeking and learning with a cognitive tutor, help provision and help seeking in interactive learning environments", paper presented at workshop held in conjunction with AIED-2001, San Antonio, TX, May.

Alpert, S.R., Singley, M.K. and Fairweather, P.G. (1999), "Deploying intelligent tutors on the web: an architecture and an example", *International Journal of Artificial Intelligence in Education*, Vol. 10, pp. 183-97.

Anderson, J.R. (1990), *Cognitive Psychology and its Implications*, 3rd ed., W.H. Freeman, San Francisco, CA, ISBN 0-7167-2085-X.

Arruarte, A., Elorriaga, J.A., Fernández-Castro, I. and Ferrero, B. (1996), "Knowledge reusability: some experiences in intelligent tutoring systems", Position Paper for ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montreal, 10 June, available at: http://advlearn.lrdc.pitt.edu/its-arch/papers/arruarte.html

Blumenthal, R., Meiskey, L., Dooley, S. and Sparks, R. (1996), "Reducing development costs with intelligent tutoring system shells", position paper for ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITS's, Montreal, 10 June, available at: http://advlearn.lrdc.pitt.edu/its-arch/papers/blumenthal.html

Brusilovsky, P. (1996), "Methods and techniques of adaptive hypermedia", *User Modelling and User-Adapted Interaction*, Vol. 6 Nos 2-3, pp. 87-129.

Brusilovsky, P. (2001), "Adaptive hypermedia", *User Modeling and User-Adapted Interaction (UMUAI)*, Vol. 11 Nos 1-2, pp. 129-58.

Chen, S.Y-H. and Ford, N. (1997), "Towards adaptive information systems: individual differences and hypermedia", *Information Research*, Vol. 3 No. 2, ISSN 1368-1613, available at: http://informationr.net/ir/3-2/paper37.html

Duval, E. (2001), "Standardized metadata for education: a status report", paper presented at World conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA) 2001, Tampere.

Felder, R.M. (1993), "Reaching the second tier: learning and teaching styles in college science education", *J. College Science Teaching*, Vol. 23 No. 5, pp. 286-90, available at: www2.ncsu.edu/unity/lockers/users/f/felder/public/Papers/Secondtier.html

Felder, R.M. and Silverman, L. (1988), "Learning and teaching styles in engineering education", *Engineering Education*, Vol. 78 No. 7, pp. 674-81.

Fischer, G. and Scharff, E. (1998), "Learning technologies in support of self-directed learning", *Journal of Interactive Media in Education*, Vol. 98 No. 4, available at: www-jime.open.ac.uk/98/4

Gehne, R., Jesshope, C.R. and Zhang, J. (2001), "Technology integrated learning environment – a web-based distance learning system", *Proceedings of IASTED International Conference 2001, Internet and Multimedia Systems and Applications, Hawaii*, ISBN 0-88986-299-0, pp. 1-6.

Houston, J.P. (1981), *Fundamentals of Learning and Memory*, 2nd ed., Academic Press, New York, NY.

Karagiannidis, C. and Sampson, D. (2001a), "An architecture for defining re-usable adaptive educational e-content", paper presented at IEEE International Conference on Advanced Learning Technologies (ICALT 01), Madison, WI.

Karagiannidis, C., Sampson, D. and Brusilovsky, P. (2001a), "Layered evaluation of adaptive and personalised educational applications and services", paper presented at 10th International Conference on Artificial Intelligence in Education (AI-ED 01), Workshop on Assessment Methods in Web-based Learning Environments & Adaptive Hypermedia, San Antonio, TX, 19-23 May.

Karagiannidis, C., Sampson, D. and Brusilovsky, P. (2001b), "Empirical evaluation of user models and user-adapted systems: a layered evaluation review", paper presented at 8th Panhellenic Conference on Informatics, Workshop on Software Usability, Nicosia, 8-10 November.

Kimble, G.A. (1961), *Hilgard and Marquis' Conditioning and Learning*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ.

Kobsa, A. (2001), "Generic user modeling systems", *User Modeling and User-Adapted Interaction*, Vol. 11, pp. 49-63, available at: www1.ics.uci.edu/~kobsa/papers/2001-UMUAI-kobsa.pdf

Kuisma, J.J. and Watson, J.A. (2002), "Scheduling and monitoring dynamic learning objects on the web", paper presented at World Conference on Educational Hypermedia and Telecommunications (ED-MEDIA 2002), Denver, CO, 24-29 June.

Mandler, J. (1984), *Stories, Scripts, and Scenes: Aspects of Schema Theory*, Erlbaum, Hillsdale, NJ.

Mayer, R.E. and Moreno, R. (2002), "Aids to computer-based multimedia learning", *Learning and Instruction*, Vol. 1, pp. 107-19.

Mullier, D.J., Hobbs, D.J. and Moore, D.J. (1998), "A web based intelligent tutoring system", *Proceedings of NETIES at Leeds Metropolitan University: An Initial Investigation into Moving the Ideas Discussed in the Previous Papers into Web Environment*, available at: www.mullier.co.uk/NETIES.htm

Murray, T. (1999), "Authoring intelligent tutoring systems: an analysis of the state of the art", *IJAIED International Journal of Artificial Intelligence in Education*, Vol. 10, pp. 98-129.

Narinen, K. (2002), "Graphical user interface for an online learning management system (LMS)", BEng thesis submitted to Turku Polytechnic, Turku.

Quinn, N. and Holland, D. (1987), *Cultural Models of Language and Thought*, Cambridge University Press, New York, NY.

Ritter, S. and Koedinger, K.R. (1995), "Towards lightweight tutoring agents", in Greer, J. (Ed.), *Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education, Washington, DC*, AACE 91-98.

Ritter, S. and Koedinger, K.R. (1996), "An architecture for plug-in tutor agents", *Journal of Artificial Intelligence in Education*, Vol. 7 Nos 3/4, pp. 315-47.

Ritter, S., Brusilovsky, P. and Medvedeva, O. (1998), "Creating more versatile intelligent learning environments with a component-based architecture", Intelligent Tutoring Systems (ITS-98) available at: www2.sis.pitt.edu/~peterb/papers/ITS98.html

Rumelhart, D.E. (1980), "Schemata: the building blocks of cognition", in Spiro, R.J., Bruce, B. and Brewer, W.F. (Eds), *Theoretical Issues in Reading and Comprehension*, Erlbaum, Hillsdale, NJ.

Sampson, D., Karagiannidis, C. and Kinshuk (2002), "Personalised learning: educational, technological and standardisation perspectives", *Interactive Educational Multimedia*, Vol. 4, (on-line journal ISSN 1576-4990), Special issue on Adaptive Educational Multimedia.

Self, J. (1999), "The defining characteristics of intelligent tutoring systems research: ITS's care, precisely", *International Journal of Artificial Intelligence in Education*, Vol. 10, pp. 350-64.

Skinner, B.F. (1958), "Teaching machines", *Science*, Vol. 128, pp. 969-77.

Stankov, S., Glavinić, V. and Rosić, M. (2000), "On knowledge representation in an intelligent tutoring system", paper presented at 4th International Conference on Intelligent Engineering Systems (INES-2000), Portorož, 17-19 September.

Suzuki, R., Hasegawa, A., Kashihara, J. and Toyaoda, J. (2001), "A navigation path planning assistant for web-based learning", paper presented at World Conference on Educational Hypermedia and Telecommunications (ED-MEDIA 2001), Tampere, 25-30 June.

Sweller, J. (1999), *Instructional Design in Technical Areas*, Australian Council for Educational Research, Camberwell.

Weber, G. and Brusilovsky, P. (2001), "ELM-ART: an adaptive versatile system for web-based instruction", *International Journal of Artificial Intelligence in Education*, Vol. 12, pp. 351-84.

Further reading

Advanced Distributed Learning (ADL) (2001), *SCORM V1.1*, Sharable Content Object Reference Model (SCORM), Version 1.1.

Advanced Distributed Learning (ADL) (2002a), *SCORM V1.2*, Sharable Content Object Reference Model (SCORM), Version 1.2.

Advanced Distributed Learning (ADL) (2002b), *SCORM (2002)*, available at: www.adlnet.org

Advanced Learning Infrastructure Consortium (ALIC) (2002), available at: www.alic.gr.jp/eng/index.htm

ARIADNE (2002), *ARIADNE Educational Metadata Recommendation*, Version 3.2., ARIADNE Foundation.

CEN/ISSS (2002a), available at: www.cenorm.be/isss/

CEN/ISSS (2002b), *CEN/ISSS Workgroup: Internationalisation of the IEEE LTSC LOM specification (2002)*, available at: www-gist.det.uvigo.es/~lanido/LOMInt/

CORBA/IIOP (2001), *Common Object Request Broker Architecture (CORBA/IIOP)*, Version 2.6, 1 December, available at: www.omg.org/

DazzlerMax (2002), *MaxIt Corporation*, DazzlerMax authoring tool, available at: www.maxit.com/daz_product_info.html

DOM L1 (2000), *Document Object Model Level 1 Specification*, 2nd ed., World Wide Web Consortium (W3C), available at: www.w3.org/TR/2000/WD-DOM-Level-1-20000929/

Dublin Core Meta-Data Initiative (DCMI) (2002), available at: http://dublincore.org/

EcmaScript (1999), *European Computer Manufacturers Association (ECMA) ECMAScript Language Specification*, 3rd ed., Standard ECMA-262, 29th September 2001, available at: www.ecma.ch/ecma1/STAND/ECMA-262.HTM

Global Knowledge (2002), *Global Knowledge*, available at: http://kp.globalknowledge.com

Grigoriadou, M., Papanikolaou, K., Kornilakis, H. and Magoulas, G. (2001), "INSPIRE: an intelligent system for personalized instruction in a remote environment", *Proceedings of the 8th International Conference on User Modelling (UM2001)*, available at: http://wwwis.win.tue.nl/ah2001/proceedings.html

Guven Smith, S. (2002), "Intelligent tutoring systems – personal notes, online hypertext tutorial", available at: www.cs.mdx.ac.uk/staffpages/serengul/ (accessed 27 May 2002).

Heift, T. and Nicholson, D. (2001), "Web delivery of adaptive and interactive language tutoring", *International Journal of Artificial Intelligence in Education*, Vol. 12, pp. 310-24.

Henze, N. and Nejdl, W. (2001), "Adaptation in open corpus hypermedia", *International Journal of Artificial Intelligence in Education*, Vol. 12, pp. 325-50.

IEEE LTSC (2002), available at: http://grouper.ieee.org/groups/ltsc/index.html

IMS CP (2001), *IMS Content Packaging Specification*, Version 1.1.2. IMS Consortium, 10 August, available at: www.imsproject.com/specifications.html

IMS Global Consortium Inc (2002), available at: www.imsproject.org

ISO/IEC JTC1/SC36 (2002), available at: http://jtc1sc36.org/

Karagiannidis, C. and Sampson, D. (2002), "Re-using adaptation logics for personalised access to educational e-content", paper presented at workshop: adaptive systems for web-based education, 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems (AH-2002), Malaga.

LIBRIX (2002), "Librix product data sheet", available at: www.maritzlearning.com/lpms.htm.

Macromedia (2002), available at: www.macromedia.com.

Mayo, M. and Mitrovic, A. (2001), "Optimising ITS behaviour with Bayesian networks and decision theory", *International Journal of Artificial Intelligence in Education (IJAIED 2001)*, Vol. 12, pp. 124-53.

Mitrovic, A. (2000), "Porting SQL-Tutor to the web", *Proceedings of the International Workshop on Adaptive an Intelligent Web-Based Educational Systems held in conjunction with ITS2000, Montreal*.

OMG/IDL (2000), *Object Management Group (OMG) Interface Definition Language (IDL)*, defined in The Common Object Request Broker (CORBA): Architecture and Specification, Version 2.2, available at: www.omg.org/

RadAuthor (2002), *Global Competency Systems*, RadAuthor, available at: www.radauthor.com/

Ritter, S. (1997), "Communication, cooperation and competition among multiple tutor agents", in Boulay, B.D. and Mizoguchi, R. (Eds), *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, IOS, Amsterdam.

Sampson, D., Karagiannidis, C. and Cardinali, F. (2001), "A personalised web-based architecture for defining re-usable adaptive educational e-content", paper presented at 8th Panhellenic Conference on Informatics, Nicosia, 8-10 November.

Stuart, D.L. *et al.* (1999), "A guide to online teaching: existing tools & projects", report commissioned by the JISC Technology Applications Programme.

Tomcat (2002), *The Jakarta Project – Apache Tomcat*, available at: http://jakarta.apache.org/tomcat/

About the authors

Jason Watson, a Senior Lecturer in the School of Information Systems, and an Online Teaching Fellow, at Queensland University of Technology, specialises in e-learning, web development and e-commerce. He is co-leader of the Centre for Information Technology Innovation (CITI) research programme "Learning Innovation in and with ICT" (LI) and Director of the 2005 FIT Teaching and Learning Large grant "Supporting explorative learning by providing collaborative problem centred e-learning environments" (COPS). Jason has published several refereed papers and sits on the review board for the Journal of Information Technology Education (JITE) and the International Journal of Learning Technology (IJLT). Jason Watson is the corresponding author and can be contacted at: ja.watson@qut.edu.au

Pervaiz K Ahmed is Director, Japanese Management Research Unit, and Head of the Centre for Enterprise Excellence. He is full Professor at the Univeristy of Wolverhampton, UK where he holds the Chair in Management. He has published numerous papers in international journals and has presented as keynote speaker at international platforms. He currently serves on the editorial advisory board of a number of international journals. He is also active in the European Foundation for Quality Management, and served as a panel judge for academic awards for four years.

Glenn Hardaker is professor of innovation and learning at the University of Huddersfield. He is also full Professor of innovation management. Research is focused on "technology led innovation & learning" in the context of: firstly, educational technologies specific to pedagogy and learning design; secondly, multicultural education and ICT; and digital supply chain management in creative industries. Glenn is editor of the international journal Campus-Wide Information Systems and editor of Multicultural Education & Technology Journal.