Education and Information Technologies 8:4, 345–368, 2003.

Learning Objects Based Framework for Self-Adaptive Learning

YACINE ATIF
College of Information Technology, UAE University, P.O. Box 15551, United Arab Emirates
E-mail: Yacine.Atif@uaeu.ac.ae

RACHID BENLAMRI and JAWAD BERRI
Etisalat College of Engineering, Department of Computer Engineering, P.O. Box 980, Sharjah, United Arab Emirates
E-mails: {benlamri;berri}@ece.ac.ae

Abstract

How learners can build their own knowledge, which is precisely tailored to their needs and background? This is precisely the question to which this paper attempts to answer by providing a framework for a flexible object-based e-learning environment. The paper recognizes that the general learner modeling alternative is an intractable problem. However, it suggests learning objects construct used as building blocks to root out individual learning deficiencies. The paper also provides an algorithm to construct individual learning routes that are adjusted to learners' profile as well as an open implementation to accommodate the integration of various learning sources.

Keywords: adaptive e-learning, multimedia material for education, learning objects, learner modeling, hypermedia courseware

1. Introduction

With recent advances in fiber optics and hardware technology, a variety of distributed multimedia applications is now becoming feasible including e-learning applications. The recent proliferation of multimedia hardware and software has made it possible for more people to produce and to distribute multimedia-based e-learning documents (Atif, 2001). However, despite the vast storehouse of published digital learning resources, it is generally agreed that education technology has not yet realized the full potential of deploying these resources. This is mainly due to the fact that the traditional mode of instruction (one-to-many lecturing), which has also been imitated by conventional education technology, cannot fully accommodate the different learning styles, strategies, and preferences of diverse learners (Manouselis and Sampson, 2002). Educators in the information technology era will have to take the role of guides by facilitating learning rather than delivering knowledge. On the other hands, today's mature learners prevail taking control of their own learning process in an active mode rather than in a massive, receiving way (Fung and Yeung, 2000). Recent progress in instructional technology (Manouselis and Sampson, 2002; Fung and Yeung, 2000; Harmse and Thomas, 2001; Teixeira *et al.*, 2002; Dufresne, 2000;

Rodriguez *et al*., 2002) has shown that new technologies have the potential to transform the learning enterprise to the benefit of learners and educators alike. In this context, the concept of learning object (LO) has emerged, and LO standardization has been receiving increased attention from the IEEE, the IMS Global Learning Consortium, the Advanced Distributed Learning (ADL) co-laboratory, and others (Rodriguez *et al*., 2002).

Amongst the developed standards are the Learning Object Metadata (LOM) (IMS GLC, 2003b) by the IEEE/IMS, the Learner Information Packaging (LIP)/Learning Object Packaging (LOP) standards developed by the IMS (IMS GLC, 2001, 2003a), and the Sharable Content Object Reference Model (SCORM) developed by ADL (2001). The main focus of most of these research efforts was on LO interoperability, aiming at facilitate the development of a standards-based e-learning industry in order to lower the overall cost and development time of e-learning products. However, there has been limited emphasis on the need for introducing adaptive learning features within the LO construct. Under the current version of these standards, the LO is treated as an opaque entity that cannot yet be adapted to learners' needs (Manouselis and Sampson, 2002; Fung and Yeung, 2000; Harmse and Thomas, 2001; Teixeira *et al*., 2002; Dufresne, 2000; Rodriguez *et al*., 2002). This paper addresses this shortcoming in the content of learning object to suit the level of learners. This is done by expanding semantically the LO to accommodate individual learner's profile and needs, and to enable dynamic generation of personalized learning routes.[1]

2. Background and Related Work

2.1. Strategic approaches to e-learning

2.1.1. Instructive vs. constructive learning The education research community advocated two separate learning strategies: the instructive model and the constructive model (Duffy and Jonassen, 1991). The instructive model simulates the instructor task in a classroom environment whereby a learner is guided through a step-by-step process towards the targeted course objectives. This model does not take into account the individual learners' differences as well as their prior knowledge or present motivation. Thus, learning systems based on this model have limited interactive capabilities and typically offer a single learning channel that is centered on the instructor. The constructive model however allows learners to rather build their own knowledge following possibly different learning paths based on the level and the background disparities of learners. Thus, learning is tailored to the continuously modified individual learner's requirements, abilities, preferences, interests and skills (Manouselis and Sampson, 2002). In this paper we adopted the constructivist approach to enable adaptive learning through the provision of instructional building blocks based on a learning objects framework.

2.1.2. Synchronous vs. asynchronous learning Due to the advances made in networking technology as well as the widespread availability of personal computers, the focus of

[1] "Learning route" and "learning path" are used interchangeably in this paper.

e-learning has shifted from delivery considerations to development considerations (Gibbons *et al.*, 2001). The above network term is used here to refer to computer network. In e-learning environments, the term network also refers to "people network" reflecting the process by which learners access learning material. This process includes two major existing e-learning delivery modes which are synchronous and asynchronous modes. Asynchronous mode of delivery brings to reality the vision that anyone can access education at anytime and from anyplace. Synchronous delivery mode requires the learner to synchronize his or her schedule with anyone else or with any other event. By this definition, attending a class, either face-to-face or through interactive TV, or even being connected to a peer group or an instructor or a tutor are synchronous learning delivery modes.

We adopted an asynchronous approach for e-learning delivery in order to deliver learning without regard to distance or time constraints to support the "anytime, anywhere self-learning" principle. Furthermore this mode of delivery increases the access to education to a pool of learners who currently may not have this access (either due to time, distance or technology constraints), and so it increases the scale of on-line learners community. This approach allows also learners to benefit from self-learning and adaptive opportunities available on-line.

2.2. Learning objects approaches

A major current focus in designing modern e-learning content is the actual concentration on efficient production of instructional components or objects which are interoperable and reusable (Najjar, 1996). With no doubt the concept of reusability becomes a key issue for new e-learning initiatives. The reusability of learning objects provides a framework that builds on past experience and creates new mechanisms for producing and exchanging knowledge. There is an actual need to discover ways for the integration of various sources of knowledge within a business or educational organization and especially to collaborate and to exchange learning objects.

Learning objects are self-contained instructional units which content accommodates heterogeneous learning sources (text, presentation, audio or video) or a combination of any of these media. The e-learning system presented in this paper promotes such generic object-based learning which has the ability to capture any learning source. Learning objects can represent small instructional components such as a course unit or an entire course that can be reused a number of times in different contexts or even an entire curriculum. However coarse grain objects reduce their reusability. In addition, learning objects are assumed to be delivered over the Internet in an open system framework free from any vendor-specific container.

The learning object design forces a certain e-pedagogy discipline in order for instruction designers to operate under a well-defined framework that prevent the design of lengthy and discursive material which may not benefit learners. The Learning Technology Standards Committee (LTSC, 2000) defines learning objects as "entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning. Examples of technology-supported learning include computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, and collaborative learning environments. Examples of learning objects include multimedia content, instructional content, learning objectives, instructional software and software tools, as well as information related to persons, organizations, or events referenced during technology supported learning" (LTSC, 2000).

In this paper, learning objects are defined broadly enough to encompass resources currently available on the Internet including multimedia resources. Our learning objects synthesize an instructional design able to capture instruction in any Internet-transferable media as well as their combination to allow instruction and practice for instance to use different media. Moreover, learning objects are inter-linked to form a network of learning resources through which learners navigate to build a personalized learning path. Hence, learning objects appear as modular building blocks, which can be easily integrated to manage e-learning content according to a specific learning strategy.

Many research works have been carried out to develop a system that adapts its learning strategies in agreement to the learner profile. In such systems, learner behavior monitoring and modeling are the most important aspects of investigation. A number of systems have used intelligent agents for user monitoring and guidance (Rickel and Johnson, 1998; Maulsby, 1993; Lieberman, 1993; El-Khouly *et al*., 1999), or as a basis for the organizational structure of the whole system (Maes *et al*., 1996; Boys, 1997; Cheikes, 1995). For these systems, the provision of personalized e-learning services is solely dependent on the intelligence of the front-end agents. On the contrary, this work embeds most of the intelligence into the learning objects themselves. This is done by adding a new dimension to the LO metadata that reflects the learner's needs. This additional metadata enables personalized learning by providing five LO functionalities described in later sections of this paper.

3. Adaptive Learning Environment

This paper proposes an approach to the design of an open adaptive learning environment in which learners dynamically select a learning route suitable to their needs and profile. The proposed environment is based on the IEEE/IMS learning object metadata (LOM) standard (LTSC/IEEE, 2001). The nature of adaptations provided by this environment are centred on the learner, and allow the LO to adapt to the evolving learner's model in terms of background, learning modalities, and learning styles. In this section we describe the enhanced adaptive LO specification, then, we overview the open system architecture for adaptive learning, and finally, we present a detailed description of an algorithm that generates a sequence of learning objects instances representing a personalized learning route within a *learning web*.

3.1. Learning object attributes

Before describing the structure of the learning object attributes, it is important to understand the functional requirements of learning objects in terms of courseware authoring, integration, interaction and media selection. When developing courseware content, the instructor may break down the subject matter into a network of concepts representing several layers of varied details and depth to achieve the instructional goals. Supporting this same process, learning objects represent small and reusable chunks of instructional media. This object-based segmentation of knowledge has been adopted in this paper to provide a constructive approach to e-learning (Bannan-Ritland *et al.*, 2001). Thus, the learning object is used as the building block to form courseware content in a process centered on the learner to "free learners from the drudgery of doing exactly similar tasks unadjusted and untailored to their individual needs" (Gibbons *et al.*, 2001) which is the case in traditional classroom instruction. A courseware structure is represented as a web of learning objects for a particular course, representing the various concepts interdependencies among learning objects. Finally, a learning route is a subset of the courseware web represented by a sequence of instance-LOs to which a particular learner get exposed during a learning session, following a particular learning style and adopting certain learning modalities.

To allow the deployment of LOs in an adaptive way, extensions to the existing IEEE/IMS LOM specification were introduced to dynamically allow LO integration, LO correlation, media selection and learner-LO interaction. Thus, the proposed metadata structure describes a multimedia rich interactive LO. Like the standard LOM, the proposed LOM includes elements such as general, rights, lifecycle, classification and annotation to describe the static features of the learning object. However, additional features were added such as educational, technical and relation to dynamically adapt the LO to learners' needs. The proposed structure of learning object attributes is depicted in Figure 1.

Learning object resources construct is rather *media-centric* which captures both tangible and intangible formats of learning. Learning resources such as text script, video, animations and images represent the static attributes of the learning object. These are the attributes which cannot be modified when reused. On the other hand, relation, technical and educational related features (LTSC/IEEE, 2001) represent the dynamic attributes which can modify some aspects of the LO when reused. Below, we describe the educational, technical and relation attributes. Metadata attribute is presented later in this section of the paper. The remaining attributes are similar to the LOM specification:

– The educational element is enhanced with features related to media selection, analogy, assessment and customization. Below we provide a description of each if these features:

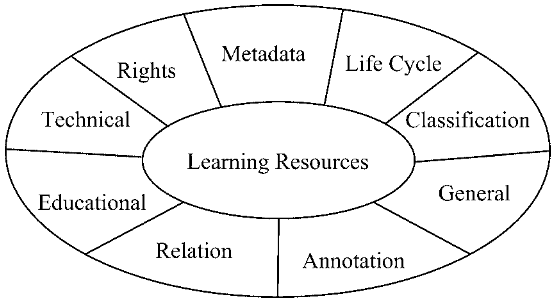

Figure 1. Learning object attributes.

  - Media selection allows a learner to customize the object to zoom on a particular media in case the learning object is a combination of multiple media.
  - Analogy facilitates learning by analogy and offers learners alternatives to comprehend further the subject imbedded in the LO.
  - Assessment enables problem-based learning and corresponds to a particular assessment strategy by which the learner can assess his understanding of the material embedded in the learning object.
  - Customization provides learners with the opportunity to augment learning content during instruction by taking their own notes. The customization assumes an authentication process for learners who may then take notes through the system at playback time to reflect their own understanding of the presented material. These notes are attached to the profile of the learner for the currently being played object.

- The technical attributes represent the synchronization and layout features describing respectively the level of synchronization involved in combining multiple media, and the actual time and space distribution of the learning media.
- Relation corresponds to the "*correlation*" feature which reflects the self-adaptability nature of the LO. As a response to a learner state, a new learning sequence of learning objects is generated to control the self-adjustment of the presented material to the learner state. This attribute contributes in self-adjusting the presented learning material based on the behavior of the learners as dictated by the constructivist approach. Different learners follow different learning routes suitable to their background level and understanding pace. Later in this section of the paper we present an algorithm that builds a learning route dynamically composed of selected learning objects, based on the learners' interaction with the system.

3.2. E-learning system design based on learning objects

This paper proposes a model for packaging learning content into learning objects and a web connectivity of learning objects through which a personalized learning route is identified automatically throughout a learning session. As shown in Figure 2, the system consists of three conceptual layers: authoring layer, LO production layer and LO composition layer. The authoring layer allows courseware authors to build LO content. The system uses industry standards for learning resource authoring and management. This ensures LO interoperability and thus allows LO export/import from/to other learning systems. The LO production layer is crucial to adapt LO content to the targeted learners. This is done by the intelligent LO composer (ILOC) which adjusts the LO content based on the LO metadata and the information provided by the learner modeller and profiler (LMP) to cater for learners' preferences and skills. The LO metadata used in this layer includes information related to general, lifecycle, rights, classification, annotation and meta-metadata for standardization purposes, and information related to educational, relation and technical LO features for customizing the learning. The educational metadata for instance, includes alternative teaching methods used in delivering the encapsulated LO content. This is used to suit learners' preferences and skills.

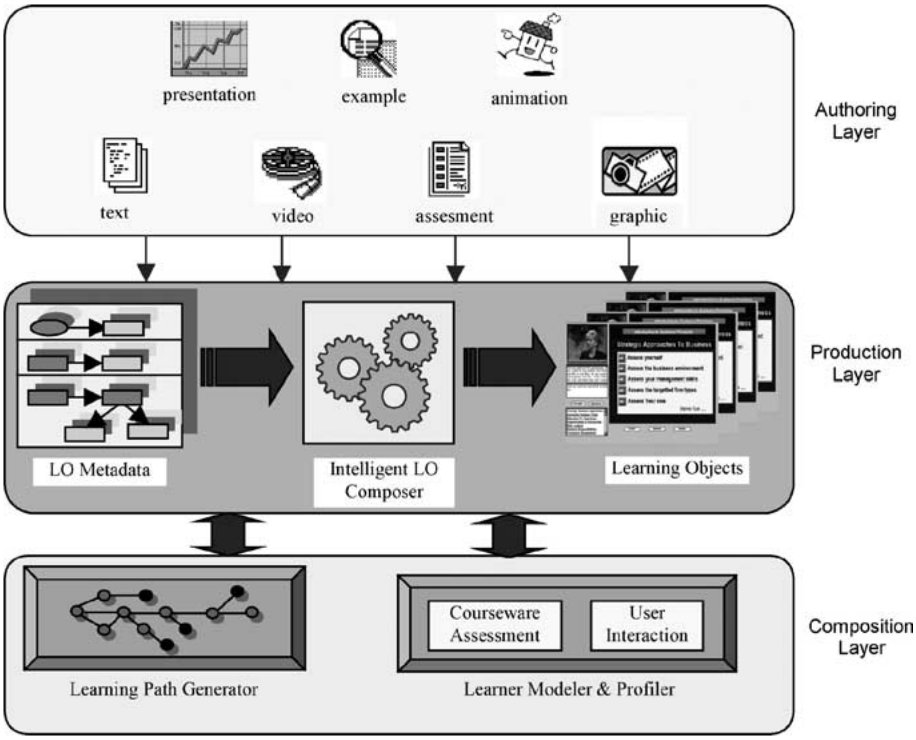

Figure 2. E-learning system design based on learning objects.

The LO composition layer consist of two components: the learner modeller and profiler (LMP), and the learning path generator algorithm. The LMP has a courseware assessment tool that allows initial overall assessment of the learner's learning styles, learning modalities, and background. The learning attributes of interest in this study are those related to whether the user learns by analogy, problem solving, detailed tasks, or by tasks which require abstract thinking. Also, the learning modalities considered in this work are related to whether they are visual, auditory or kinaesthetic learners. These learning attributes are represented by variables which measure the learner's preferences.

3.3. Learning process using LOs

The LO metadata attribute shown in Figure 1 enable personalized learning by providing five LO functionalities. These are LO sequencing, LO content and structure, LO presentation, LO navigation support and LO interactivity. Below, we describe each of these LO functionalities and their contribution to enable adaptive learning.

3.3.1. LOs sequencing The Learning Objects Metadata Working Group's (a working group of the Learning Technology Standards Committee (LTSC, 2000)) aims at promoting adaptive e-learning systems to enable computer agents to automatically and dynamically compose personalized lessons for an individual learner. To achieve this objective, this working group stated that "instructional design should not be structured in the traditional sequential format whereby all learners get the same instruction regardless of their individual needs and background". Alternatively, learning objects should be invoked dynamically to form a learning path that is suitable to root out the learning deficiencies of individual learners. A huge focus is currently taking place to build personalized learning. It is essential to view any implementation of personalized learning as a joint cognitive system involving a variety of learning models. For instance, a *serialist* courseware author may feel more enthusiastic about a tightly constrained system designed on the building blocks metaphor, while a *holist* author may be motivated by a loosely constrained system that allows zooming in and out of fine grained details. Similarly a *pragmatist* courseware author may prefer a focus on practical applications while a *theorist* author may prefer logical analysis (Patel and Kinshuk, 1998). The proposed system attempts to address these various learning styles bearing in mind that it is outside the scope of this paper to consider each learning style in great details.

Different techniques can be used to track learners' behaviour in order to invoke the appropriate LO sequence that provides personalized learning content. As learners interact with the e-learning content, results could be communicated to the e-learning system which may then adapt the content based on the learner information. For example, learners might be sent to different places in the content based on user-initiated request for clarification of prerequisite knowledge, or user requests for supportive knowledge expressed in terms of examples, case studies or procedural information. In the proposed system, each learning object has semantic connections with other objects. Different users navigate across the learning web composing the learning objects interconnectivity following different paths. The learning path-building process, which contains the sequence of objects exposed to a learner, is performed dynamically based on the user's requests and profile. While replying to user requests, learning objects experienced so far are dynamically removed from the learning path. A detailed description on how to sequence learning objects in order to build personalized learning based on the above mentioned learning styles is given below.

Each request for a prerequisite knowledge is formulated in the form of a hyperlink embedded in the text component of a LO which links that learning object to a prerequisite learning object. Also, the learner, on request, can explore a relevant list of predefined prerequisite learning objects related to the one currently explored, if any. This list is dynamically updated based on the learning objects visited so far. Finally, alternative learning styles are exhibited through the use of additional learning objects such as examples, case studies, and procedural information to provide personalized learning. These options give learners, who have different intellectual capabilities, the flexibility to choose a suitable learning path instead of adopting a rigid one. This allows the learner to make an informed decision regarding where to proceed in the material. The possible sequences that might be invoked from a LO are expert-defined and exhibit different learning strategies. However, these are not necessary all enabled at a given time, but depends on the learner's skills, prerequisite knowledge, learning style and courseware navigation history. The system dynamically adjusts relevant sequences based on the learner's profile.

3.3.2. LOs structure This section describes the educational effectiveness and pedagogic features of the LO. It consists of a sequence of learning tasks to accomplish the goals and objectives set up by the courseware author for a good understanding of concepts presented in the LO. These are combination of learning resources similar to those listed in the educational component of the LOM specification (LTSC/IEEE, 2001), and can be: slides, examples, questions, problems, simulations, case studies, experiments, diagrams, graphs and so forth. These are structured in a way to allow different learning styles at different levels. Depending on the learner's profile and preferences, the LO content can describe the learned concepts in many ways, such as, abstract form (generalized learning), through a problem solving (problem-based learning), or using a case study (learning by analogy) and so forth.

3.3.3. LOs presentation LO presentation describes the way individualized learning materials imbedded into the LO are dynamically presented to the learner. Multimedia contributes further to learning when instructional designers use the most effective medium to present specific information. Hence, there is a need for instructional designer to map a learning content to an appropriate media. A number of empirical studies suggest how to select specific media or a combination of media for successfully presenting specific kinds of learning content as summarized in Table 1.

The content-to-media mapping shown in Table 1 has been confirmed through empirical experiments to provide the best media allocation for learning content (Najjar, 1996). Assembly instructions are best comprehended when an assembly task is presented using a combination of illustrations and text highlighting the major steps. Procedural information for operating a particular device for instance, appears to be more helpful for learners to acquire when a combination of animation or video and text is presented to learners. For problem-based learning, an animation with verbal narration was shown to be effective. For instance, solving a mathematical equation may be better illustrated through a graphical illustration. Pictures increase recognition accuracy especially when combined with text to drive the learner to focus on specific features of the pictures. Sound appears to be an effective way to communicate. For instance, in learning a particular foreign language, it would be more helpful for a learner to hear the words. But some words are context dependent and the context may be better understood if shown through video. And to help the language-learner further, a textual version of the words' phonetic would reinforce the learning process of such verbal information. Finally, recalling story details would be more effective with a video or a soundtrack. The e-learning system presented in this paper pro-

Table 1. Media allocation

Learning content	Media
Assembly instructions	Text with supportive pictures
Procedural information	Text with animation or video
Problem solving	Animation with explanatory verbal narration
Recognition	Pictures with text or verbal narration
Verbal	Sound or video and text
Story details	Video with a soundtrack

vides opportunities to map knowledge imbedded in LOs in any of the above formats or a combination of the above formats.

3.3.4. LO navigation Different LO's have different navigation alternatives, depending on their type, role, content and structure. Here we describe the possible ways of navigating within a LO. For instance, a learner starting a problem solving LO is recommended to go through all problem solving steps, however, it is not recommended to explore all alternatives in a LO consisting of a number of examples/case studies describing the same concept. By doing so, the system guides learners implicitly and leaves the choice of the next knowledge item to be learned and next problem to be solved, to them.

3.3.5. LO interactivity LO interactivity is an important aspect in the learning process. Degree of interactivity may differ from one LO to another depending on its type and role. The system allows learners to interact with most LOs, and especially with those LOs related to problem solving, questionnaires and self-assessment. Learner's responses are saved into his/her profile and may be used for personalization purposes and future guidance. The system also allows learners to take notes which are saved in their profile along with a reference to the relevant LO. Personal notes allow learners to either enhance their understanding of the learned concepts by using their own words, or to signal misunderstood concepts for future clarification.

3.4. Learning web

A learning web represents a particular courseware designed by the course author. An example of a learning web is shown in Figure 3. The learning web is constructed by the courseware author simply by identifying the sequence of learning objects references which participate in the courseware. These learning objects are *recommended* objects in the sense

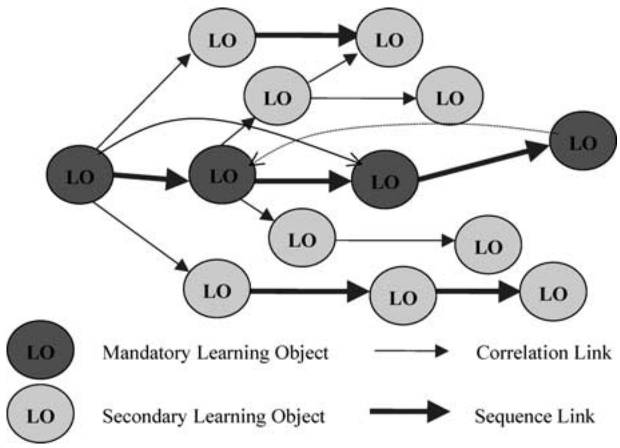

Figure 3. Learning web.

that the learner should normally visit them to fulfill the courseware understanding requirements. The recommended learning objects are the objects of the recommended learning path, i.e. the learner who does not posses prerequisite knowledge should be exposed to those objects. For the recommended path, the courseware author considers each recommended object individually to identify its correlative sequence of references to secondary objects. The process of building a correlative sequence is re-iterated on each secondary object. Secondary objects are objects of the learning web which may be added to the learning path dynamically based on the learner interactions with the learning objects.

The above learning web structure represents a body of knowledge, which is highly structured. Full comprehension of a topic may be dependent on the understanding of one or several other concepts. In a properly organized course, a particular concept is presented only after all concepts, on which it depends, have already been presented. Furthermore, a competent instructor will not proceed before insuring that the majority of the students have mastered or at least have been exposed to the prerequisite concepts otherwise, the instructional process will not be very effective. However, since we are dealing with individuals and not a group of learners, there should be some flexibility in presenting the course material to meet personal abilities of learners. All this leads us to the conclusion that the organization of knowledge within a subject matter has the form of a directed graph, not unlike the PERT charts used in project management. The graph structure will give learners the choice to either follow the recommended leaning path, or to jump to advanced concepts. While jumping to an advanced concept, the leaner is reminded with a list of prerequisite learning objects for possible review.

While this paper focuses on learning objects as an atomic learning structure, a future work considers a higher-level structure which is a courseware object which structure is similar to a learning web. Learning objects constitute then the building-blocks for the courseware objects. The objective of such future work is to allow learning objects reuse as well as courseware objects reuse. This multi-layer learning reuse would provide coarse and fine-grain reuse. As mentioned earlier in this paper coarse grain learning objects such as courseware objects reduces their reusability and that is why we provide a finer-grain reuse at the level of learning objects.

3.5. Learning path generator algorithm

The learning web represents the algorithm's search space for a learning path. The entities of the learning web are references to learning objects. These are entities through which the learning path passes. The mandatory learning objects must be in the learning path. The algorithm requires the cooperation of further secondary learning objects to find a self-adjusted learning path over the learning web. Initially, a learner initiates the learning process by requesting a courseware represented by its associated learning web. This request triggers the search for a learning path. Thus, a learning web forms the input of the self-adaptive e-learning algorithm which goal is to define a learning path for each learner. The learning path gets updated dynamically throughout a learning session when a learner invokes a correlative learning sequence. A correlation is triggered by the learner

1. While ($P \neq \emptyset$)	// As long as the target objects have not been visited
2. t_i = Invoke (P.First)	// Retrieve the object at the head of the targets list and // create a local copy of the object
3. $T_i = t_i$.Correlation	// Retrieve the sequences of correlative paths. T_{i} refers to // all the learning objects in the correlative sequences of t_i
4. $T_i^{(1)} \leftarrow T_i \backslash (T_i \cap E_p)$	// Remove the excluded learning objects
5. $T_i^{(2)} \leftarrow$ ApplyCost ($T_i^{(1)}$)	// Remove the correlations which violates the cost constraints
6. $T_i^{(3)} \leftarrow T_i^{(2)} \backslash (T_i^{(2)} \cap R)$	// Remove the visited correlative objects to avoid cycles

7. $t_i^{(1)}$.Correlation $\leftarrow T_i^{(3)}$	// Update the learning object t_i with the correlations in $T_i^{(3)}$
	// i.e. the empty correlations as a result of the
	// above transformations are not proposed to the learner

8. Play ($t_i^{(1)}$)	// Play the learning object t_i and display the correlations in $T_i^{(3)}$
9. WHILE (NOT EndPlayback($t_i^{(1)}$))	// As long as $t_i^{(1)}$ did not terminate
10. SWITCH (Event($t_i^{(1)}$))	// Listen to events
a. CASE Correlation	// A correlation has occurred

i. $CS_i \leftarrow$ Retrieve ($T_i^{(3)}$)	// Retrieve the activated correlation i.e. retrieve the
	// path of objects' references associated with the
	// activated hyperlink

ii. $P^{(1)} \leftarrow$ Insert (CS_i, P)	// CS_i is inserted at the head of P
iii. Append (t_i, R)	// t_i is appended to the tail of R
iv. GOTO 1	// To play the first object in the correlative sequence
b. CASE Backward	// The learner invoked the previously played object
i. $t_j \leftarrow R$.Last	// Retrieve the previously played object
ii. $T_j \leftarrow t_j$.Correlation	// Retrieve the correlative sequences in t_j
iii. $P^{(1)} \leftarrow$ Remove (T_j, P)	// Remove t_j's correlation in P if any
iv. $P^{(2)} \leftarrow$ Insert (t_j, $P^{(1)}$)	// Insert back t_j at the head of P
v. GOTO 1	
11. ENDCASE	
12. ENDWHILE (9)	
13. $P^{(1)} \leftarrow$ Remove ($t_i^{(1)}$, P)	// Remove the played learning object t_i from P
14. Append (t_i, R)	// Add t_i to the tail of R
15. ENDWHILE (1)	

Figure 4. Learning path generation algorithm. "\" means set subtraction.

in hope of clarifying some misunderstood concepts in the current learning object. As a result, the sequence of objects associated with the current learning object correlation become mandatory objects and are scheduled for playback. This type of navigation provides the self-adaptive dimension of the e-learning system proposed in this paper. As shown earlier, learning objects construct features a correlation attribute that represent the alternative sequence of learning objects to which the learner is to be exposed should he lacks some prerequisite concepts. When no interaction occurs during the playback of a learning object, the transition to the next learning object is dictated by the current sequence of mandatory objects.

Table 2. Algorithm's data structure

Label	Legend
P	A sequence of references to mandatory learning objects representing the targeted learning concepts initially containing the mandatory learning objects for this courseware ($P = [p_1, p_2, \ldots, p_n]$).
R	The actual learning path for the current courseware object containing initially an empty set ($R = []$) but will be updated during a learning session by the visited learning objects and their sequence.
E_P	Path constraints; i.e. a list of learning objects to be excluded from the learning path.
C	Cost constraints; i.e. constraints such as the number of learning objects on the path or the maximum time the learner can afford to allocate to a learning session.
D	Accumulated cost along the learning path.

The learning-path planning process complexity is cut down by *pruning* the alternative correlations that do not satisfy some constraints. For example, a learner can specify not to have certain learning objects on the path, because for instance he is aware of the corresponding concepts. This can be stated by indicating the identity of each individual learning object or even by pointing out a courseware. Another constraint that can be specified is the maximum time the learner is going to spend for a learning session. This unique feature self-adjusts the sequence of the presented learning objects based on the time a learner can afford to allocate to a learning session. The time attribute corresponding to each learning object is calculated based on the length of the learning media (i.e. video-clip time). Other constraints can be the number of intervening objects.

The algorithm described in Figure 4 is event-driven where events are triggered whenever a learning object is to be invoked. Learning objects are invoked remotely since they are assumed to be distributed in the learning network of an institution. Each invocation results in the construction of a local copy of the learning object which is then inserted in the learning path. An Object Request Broker assumes the responsibility of fetching remote learning objects. Once a learning path has been constructed, it then represents the learner's version of the courseware customized to his background. The learning path is then stored in the learner profile for the particular courseware he initially requested to attend.

The e-learning system architecture is based on a client/server framework where the server's role is to fetch the invoked learning objects by the client application. A 5-tuple data-structure is maintained by the client application $\{R, P, E_P, C, D\}$ which are described in Table 2. The event-driven algorithm shown in Figure 4 recognizes three types of events that can occur during the playback session of a learning object: correlation, backward move and forward move. Correlation move occurs when the user invokes the playback of a correlative sequence which results in inserting the actual objects of the correlative sequence in the learning path to prepare their playback. Backward event occurs when the learner moves backward in the learning path constructed so far. This move results in restoring the learning path to the state in which it was prior to playing the current object. Forward event corresponds to a move to the next learning object to play in the current learning sequence of mandatory learning objects.

The algorithm starts by invoking the first learning object in the learning sequence P. A local copy of the invoked object is created and inserted in the learning path, and then its media content is visualized. If the learner clicks on a hyperlink inserted within a learning object during its media playback, a correlation event has then occurred. The objects in the corresponding correlative path are then scheduled for playback and their references are inserted in the list of mandatory objects. Learning objects are displayed to the learner as long as the following constraints are satisfied:

(1) The learning objects should not be in the path constraint E_P.
(2) The learning object has not already been visited.
(3) The learning object does not violate the cost or time constraints function C.

The algorithm analyzes each learning object prior to its playback to show to the learner the appropriate correlations that satisfy the time and cost constraints and do not lead to a cycle. These correlations are reflected in our case through hyperlinks in the text script area of a learning object. The hyperlinks will not be active if the associated concept or sequence does not satisfy the abovementioned constraints.

3.6. *Example*

The illustration presented in this section traces the algorithm in a real-life context. The algorithm is applied to a fragment of a course in data structures using the C++ language (Dale and Teague, 2001). Figure 5 shows the resulting graph. The graph exhibits the course mandatory and secondary learning objects, the sequence and correlation links as defined by the courseware author. An example of a learner's specific learning route is illustrated by numbered transitions which will be traced by the algorithm given above.

Transition 0. The set P of mandatory learning objects forms the backbone of the courseware; Any learner must be exposed to the exhaustive sequence of learning objects in P. Initially, P includes the sequence: {SLS, QLS, ULL, SLL, CLL, LLA}. When the learner invokes the course, he will be exposed to the first mandatory learning object SLS. The system is said to be in the initial stage with the following data structure status:

Transition 0. Initial state	
$P = \{$SLS, QLS, ULL, SLL, CLL, LLA$\}$	mandatory objects
$t_0 = \{$SLS$\}$	object to playback
$T_0 = \{$ADT, CCT$\}$	correlative sequence of t_0
$R = \{\}$	visited LOs

Transition 1. During navigation within the learning object SLS, the system is listening to any event that can be triggered by the learner. Transition 1 (Figure 5) shows that a correlation occurs, the learner invokes the secondary learning object ADT. The system

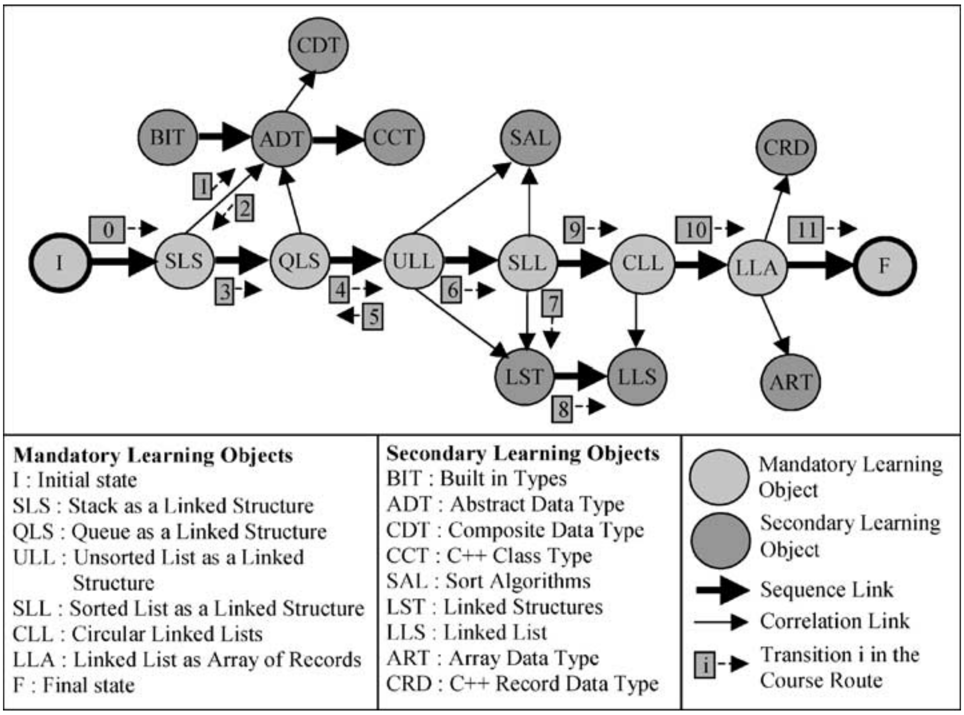

Figure 5. Example of a courseware and learning route.

time-stamps the playback of SLS at that time (θ) in order to resume it again. This is done only for the objects in the initial set P of mandatory objects, to ensure a full coverage by the learner. Then, the system pushes the correlative path of SLS, that is (ADT, CCT), in P and activates ADT for navigation. The data structure is updated accordingly as in the following:

Transition 1. Correlation occurred: *the correlation ADT is invoked*
$P = \{\text{ADT, CCT, SLS}(\theta), \text{QLS, ULL, SLL, CLL, LLA}\}$
$t_1 = \{\text{ADT}\}$
$T_1 = \{\text{CDT}\}$
$R = \{\}$

Transition 2. During navigation within ADT, the learner invokes a backward event to the previously played object. Learning object SLS is resumed for navigation from time θ. ADT is added to R – that is the visited learning objects set. Note that T_2 contains no longer ADT since it is a correlative object that has been visited and will not be proposed to the learner in the rest of the current courseware. This feature of the algorithm is one of the main ingredients that make the system adaptive to the learner. In fact, correlative objects that have been visited by the learner do not need to be presented all over again.

Transition 2. Backward move occurred: *the previously played object is invoked*
$P = \{\text{SLS}(\theta), \text{QLS}, \text{ULL}, \text{SLL}, \text{CLL}, \text{LLA}\}$
$t_2 = \{\text{SLS}(\theta)\}$
$T_2 = \{\}$
$R = \{\text{ADT}\}$

Transition 3. No event occurred, therefore SLS is completed and the next mandatory object QLS is invoked. Note that ADT is not in T_3 since it has already been visited.

Transition 3. No event occurred
$P = \{\text{QLS}, \text{ULL}, \text{SLL}, \text{CLL}, \text{LLA}\}$
$t_3 = \{\text{QLS}\}$
$T_3 = \{\}$
$R = \{\text{ADT}, \text{SLS}\}$

Transition 4. No event occurred, therefore QLS is completed and the next mandatory object ULL is invoked:

Transition 4. No event occurred
$P = \{\text{ULL}, \text{SLL}, \text{CLL}, \text{LLA}\}$
$t_4 = \{\text{ULL}\}$
$T_4 = \{\text{SAL}, \text{LST}\}$
$R = \{\text{ADT}, \text{SLS}, \text{QLS}\}$

Transition 5. During navigation within ULL, the learner invokes a backward event to the previously played object. The system time-stamps the playback of ULL and invokes QLS that is removed from R and added to the list P of learning objects:

Transition 5. Backward occurred: *the previously played object is invoked*
$P = \{\text{QLS}, \text{ULL}(\theta), \text{SLL}, \text{CLL}, \text{LLA}\}$
$t_5 = \{\text{QLS}\}$
$T_5 = \{\}$
$R = \{\text{ADT}, \text{SLS}\}$

Transition 6. During navigation within QLS (in transition 5) no event occurred, thus QLS is completed and the system moves forward to the next LO in the list P, that is ULL at time θ. Again ULL is completed and the system moves forward to the next LO, that is SLL. Note that the set of correlative objects T_6 includes SAL and LST since they have not been visited during the playback of ULL.

Transition 6
$P = \{\text{SLL, CLL, LLA}\}$
$t_6 = \{\text{SLL}\}$
$T_6 = \{\text{SAL, LST}\}$
$R = \{\text{ADT, SLS, QLS, ULL}\}$

Transition 7. During the navigation within SLL, a correlation occurs, the learner invokes the secondary learning object LST. The system time-stamps the playback of SLL and pushes the correlative path of SLL, that is (LST, LLS), in P and activates LST for navigation. The data structure state is the following:

Transition 7. Correlation occurred: *the correlation LST is invoked*
$P = \{\text{LST, LLS, SLL}(\theta)\text{, CLL, LLA}\}$
$t_7 = \{\text{LST}\}$
$T_7 = \{\}$
$R = \{\text{ADT, SLS, QLS, ULL}\}$

Transition 8. The learner invokes no backward event (in transition 7); hence LST is terminated, then LLS is invoked and completed.

Transition 8
$P = \{\text{LLS, SLL}(\theta)\text{, CLL, LLA}\}$
$t_8 = \{\text{LLS}\}$
$T_8 = \{\}$
$R = \{\text{ADT, SLS, QLS, ULL, LST}\}$

Transition 9. No event occurred, therefore the sequence of LO in P is followed: SLL is resumed from time θ and completed and the next mandatory object CLL is invoked. Note that the correlative object LLS is no longer in T_9 since it is in R (the set of visited LOs).

Transition 9.
$P = \{\text{CLL, LLA}\}$
$t_9 = \{\text{CLL}\}$
$T_9 = \{\}$
$R = \{\text{ADT, SLS, QLS, ULL, LST, LLS, SLL}\}$

Transition 10. No event occurred, therefore CLL is completed and the next mandatory object LLA is invoked.

Transition 10
$P = \{\text{LLA}\}$
$t_{10} = \{\text{LLA}\}$
$T_{10} = \{\text{CRD, ART}\}$
$R = \{\text{ADT, SLS, QLS, ULL, LST, LLS, SLL, CLL}\}$

Transition 11. This is the transition to the final state. Once LLA is completed, the set of mandatory objects P is emptied indicating the final state for the algorithm. The set R contains the sequence of the LO visited by the learner, this forms the basis of any assessment procedure.

Transition 11. Final state
$P = \{\}$
$t_{11} = \{\}$
$T_{11} = \{\}$
$R = \{\text{ADT, SLS, QLS, ULL, LST, LLS, SLL, CLL, LLA}\}$

The example above shows the traversing of the courseware for a particular learner who followed a specific learning route in the associated graph (Figure 5). The final state of the traversed learning objects and the learning route that is generated accordingly, is an instantiation of the initial graph in Figure 5 for that particular learner; another learner will interact differently with the same courseware generating therefore another instance of the same initial graph. Figure 6 shows the updated graph after complete traversing. It shows more specifically the actual path that has been traversed by the learner and, as a consequence, the updated correlations in the learning objects. For example, the initial correlation that holds between learning objects QLS and ADT is not present because it has been removed due to the prior traversing of ADT in transition 1.

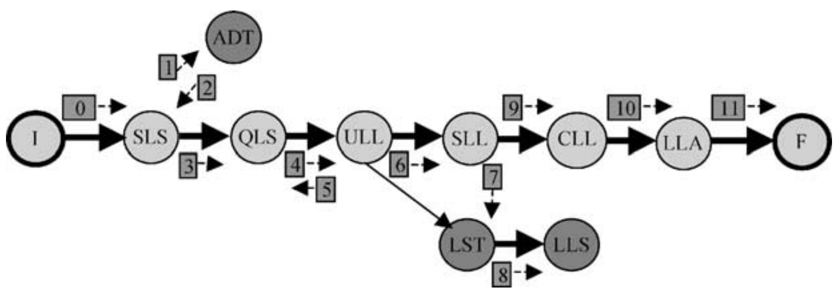

Figure 6. The updated graph after traversing LOs in the learning path.

4. System Architecture and Implementation

To make the learning web and the learning-path building process a reality, the implementation should ensure the three following requirements which are common to open system standards: interoperability, portability and integration. Interoperability means that two systems can work with each other through well-defined interfaces. This is a crucial requirement for Internet-based applications as users may build systems from diverse software development tools and may use heterogeneous computers and yet, they need a form of communication among their systems. Portability means that an application should be decoupled from a particular computing environment. Integration requires minimum efforts in using a particular system. To satisfy these requirements, we have chosen OMG's CORBA to implement the learning web presented in this paper. CORBA has been emerging as a standard for implementing and deploying distributed applications. Next, we reveal the learning object architecture to show how existing learning resources can be integrated in the learning web. Then, we present a CORBA based system architecture of the learning web followed by a description of the implementation of the learning path generator algorithm.

4.1. Learning object architecture

Successful e-learning systems are based on two major principles: modularity and abstraction. While modularity divides the system into self-contained modules or objects, abstraction separates the description of these objects from their actual implementation. E-learning system architectures based on these design principles are scalable and flexible allowing easy integration of new learning resources.

Figure 7 shows how this integration process can be fine-tuned to embody exiting learning resources and thus to become a learning object. The multi-layered architecture shown in Figure 7 divides an e-learning system into three distinctive layers:

- The interface layer exposes learning services to external entities. In this layer, learning attributes are identified and added to their related group of services.
- The learning object implementation layer contains a processing methodology of the advertised learning resources. It includes the knowledge rules and control functions which are embedded in learning objects.
- The learning object related data layer focuses on the institution learning-resources data management implemented using database servers from possibly different vendors. ODBC and JDBC drivers allow access to these heterogeneous database systems.

The open architecture shown in Figure 7 allows easy encapsulation and integration of learning services into learning objects from existing e-learning systems. This integration could be achieved by defining additional interfaces for the new services and creating the implementing objects for these services along with the required data sources.

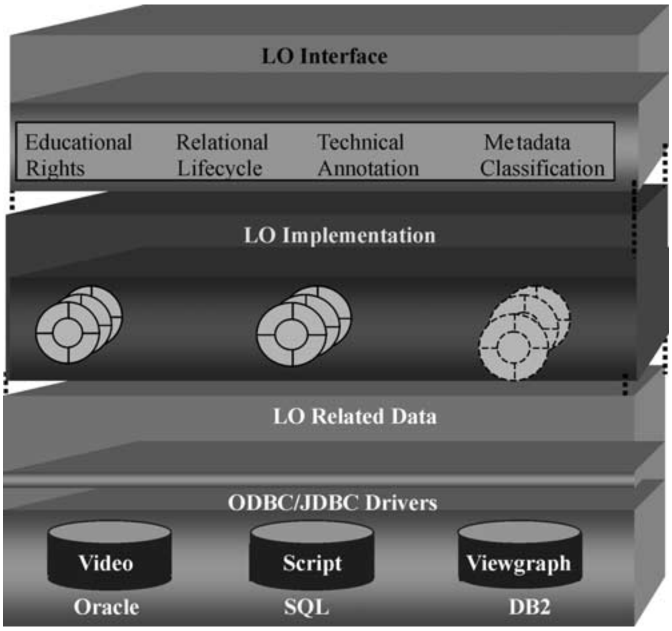

Figure 7. LO system architecture.

4.2. Learning web architecture

In the implementation of the learning web, LOs are servers as well as clients. As servers, LOs expose a remote object through its interface. LO clients can invoke the exposed-object methods remotely. LOs also become clients when they invoke other LOs' methods. As discussed earlier, the interface is a purely declarative component that hides the implementation details. This is a deliberate strategy to facilitate interoperability and software integration.

Figure 8 depicts the basic system architecture of the learning web. Based on the architecture shown in Figure 7, an LO advertises its interface of services by *plugging* it to an Object Request Broker (ORB) such as CORBA, which provides the inter-LOs interoperable connectivity. The learning web is implemented as a middleware that allows remotely located LO objects to communicate with each other. The ORB allows LOs to find each other automatically on the learning web using their individual IDs, which are stored in each LO's database. The institution where LOs are located runs also a web server so that other institutions' client objects can remotely invoke its learning resources.

To accommodate the performance requirements particularly when video information is invoked remotely, a video-server may also be required. We have used in our implementation Real-Video server to stream the video across the network. However the streaming technology is transparent to the architecture.

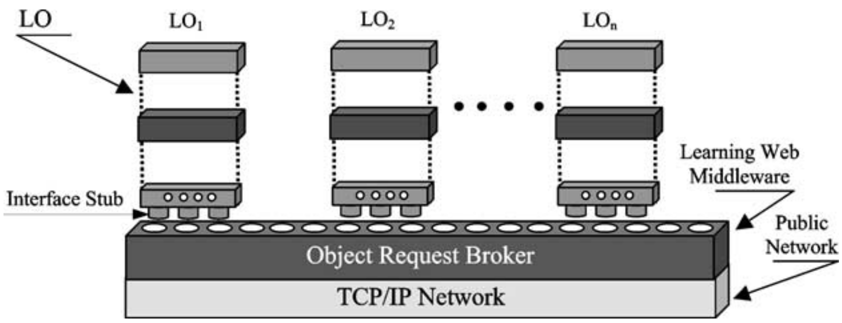

Figure 8. Learning web architecture.

4.3. Learning-path generation

A learning object may be invoked many times simultaneously as it may be involved in the construction process of multiple learning routes. Learning Object invocation is performed in the form of remote method invocation. For each invocation session, the invoked learning object spawns a thread of the requested method. Multiple threads may represent the same learning object but involved in different learning routes which represent different learners. Each thread has its own state-related information which shared section-part has been serialized to avoid any critical section violation problem. CORBA provides such a multithreading facility allowing a learning object to simultaneously receive and process multiple methods' invocations at the same time.

As described in the learning-route construction algorithm, learning objects communicate with each other through message exchanges in a client/server setup. To do this efficiently, threads are spawned at the invoking host to "bind" to the neighboring invoked learning object server and then remotely call the requested method where the arguments of that method represent the exchanged message. Initially, the learner selects a course advertised courses by his institution. The course selection induces a list of targeted mandatory learning objects references or IDs. This forms the data structure referred to in the algorithm by P. The system user interface allows the learner.

The first targeted reference is first fetched which results in creating a local copy of the learning object. Subsequent objects are invoked automatically based on the learners profile as shown in the algorithm. The data structure referred to by R in the algorithm is constructed at this stage. Correlative alternatives as well as constraints are also considered at this stage. Note that the learning material themselves embedded in the learning object are not downloaded locally but will be invoked when needed remotely from the database or the video-servers of the site hosting the invoked object. Only learner-adaptive information is constructed in the local copy of the object. This includes the correlative alternatives as well as the preset constraints.

In case video component is part of the attributes of an LO, playback of the locally formed learning object starts when the remote video-server streams the first few frames of the video while the web server downloads the other media components. During a learning session, an event will drive the algorithm towards another learning object which is invoked in a

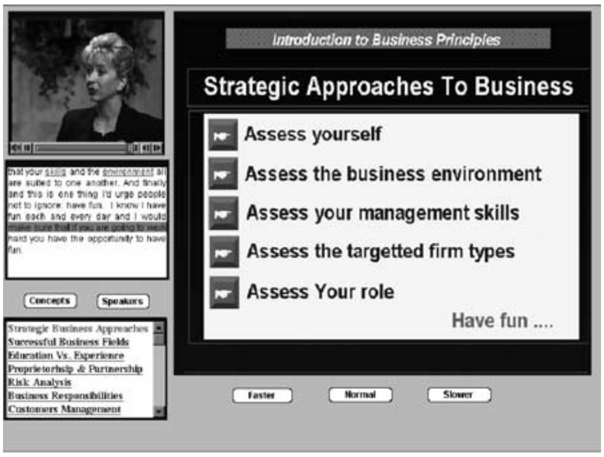

Figure 9. Learning object presentation.

similar way as previous object. Each invoked learning object is inserted in a learning route data structure (referred in the algorithm by R). This learning route can at anytime be stored for tracing or tracking the learning path of a learner as well as for resumption of paused learning session.

4.4. Learning object presentation

Our implementation of a learning object makes use of a range of multimedia information. Each learning object is presented to the learner in the form of a composite document which integrates a video script (showing for instance the talking head of the lecturer or a particular dynamic process such as a chemical experiment in progress). A sequence of images (representing related viewgraphs or slides) and an animated text-script are synchronously displayed during the video playback. Figure 9 illustrates the resulting presentation of our learning object. When a learning object is invoked, its corresponding video-script is played concurrently with the display of a sequence of images at specific time-slots and also a text transcript to help learners who have hearing impairments or difficulties to understand the lecturer accent to read the speech of the lecturer. Slides presentation instants and durations are inserted within the video timeline to be displayed at predefined time frames. The text component of a concept represents the textual version of the speech in the video. This text component automatically scrolls down during the video playback so that the appropriate speaker notes are displayed when the related video content is reached. Text acquisition is done automatically by the course author using a speech-recognition software.

The above represents media selection choices to construct our learning object with regard to our local educational environment. Other implementations of learning objects may proceed differently, but however preserving a common interface as explained earlier in Figures 7 and 8. Also learning objects presentation style and design considerations are left to the institution adopted standards. Our implementation suggests Multimedia documents forming pre-orchestrated course notes. Using their web browser such as Netscape or Explorer, learners remotely access the course presentation system to view a courseware, which triggers the corresponding learning objects path-construction process. Pre-orchestrated multimedia data refers to stored data for which the play-out scripts have already been specified at the time of authoring and storage. The system includes both a playback module for learners to view a course and a course-authoring module for instructors to design learning objects with the required standard interface.

5. Conclusion

Considerable efforts have been made in the e-learning community to standardize learning objects as knowledge building blocks. Few of these efforts however involve the learner's background, level, and learning style in the LO construct. This paper attempts to respond to the demand for self-adaptive e-learning systems by providing a framework which embeds learners' singularities in the LO description as well as in the learning process. Accordingly, the LO description is extended to include features such as educational, technical and relation attributes. The paper also suggests an algorithm to guide learners towards a customized learning route. The implementation of the presented e-learning framework further provides an open learning environment to allow easy integration of existing learning sources which contributes in expanding the volume of the learning web to offer enriching learning experiences.

References

Advanced Distributed Learning – SCORM, Version 1.2 (2001); *www.adlnet.org*

Atif, Y. (2001) Real-time network resources allocation in distributed multimedia information systems. In *Proc. of IEEE GLOBCOM*, San Antonio, Texas, USA.

Bannan-Ritland, B., Dabbagh, N., and Murphy, K. (2001) Learning object systems as constructivist learning environments: Related assumptions, theories and applications. In *The Instructional Use of Learning Objects*, D. A. Wiley (ed.). Association for Educational Communications and Technology, Bloomington, IN.

Boys, G. A. (1997) Software agents for cooperative learning. In *Software Agent*, J. M. Bradshaw (ed.). AAAI Press/MIT Press.

Cheikes, B. A. (1995) GIA: An agent based architecture for intelligent tutoring systems. In *Proceedings of CIKM Workshop on Intelligent Information Agents*, Baltimore, MA.

Dale, N. and Teague, D. (2001) *C++ Plus Data Structures* (2nd edition), Jones and Barlett (eds.).

Duffy, T. M. and Jonassen, D. H. (1991) Constructivism: New implications for educational technology. *Educational Technology*, **31**(5), 7–12.

Dufresne, A. (2000) Model of an adaptive support interface for distance learning. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems ITS'2000*, Montreal, Canada, June, pp. 334–343.

El-Khouly, M. M., Far, B. H., and Koono, Z. (1999) A multiagent Internet based tutoring system (I-CALT) for teaching computer programming languages. In *Advanced Research in Computers and Communication in Education*, G. Cummings *et al.* (eds.). IOS Press.

Fung A. C. W. and Yeung, J. C. F. (2000) An object model for a web-based adaptive educational system. In *Proceedings of the IFIP International Conference on Educational Use of Technologies (ICEUT'2000)*, China, August.

Gibbons, A. S., Nelson, J., and Richards, R. (2001) The nature and origin of instructional objects. In *The Instructional Use of Learning Objects*, D. A. Wiley (ed.). Association for Educational Communications and Technology, Bloomington, IN.

Harmse, R. and Thomas, T. (2001) Towards a conceptual UML model for tracking a learner's progress in an outcomes-based environment. In *Proceedings of the South African Institute of Computer Scientists and Information Technologists Annual Conference*, Pretoria, South Africa, September, pp. 57–65.

IMS Global Learning Consortium. (2001) IMS Content Packaging Specification; *http://www.imsproject.org/content/packaging/index.html*

IMS Global Learning Consortium. (2003a) IMS Learner Information Package Specification; *http://www.imsglobal.org/profiles/index.html*

IMS Global Learning Consortium. (2003b) *IMS Learning Resource Meta-Data Information Model*, Version 1.2.2; *http://www.imsglobal.org/metadata/imsmdv1p2p2/imsmd_infov1p2p2.html*

Lieberman, H. (1993) Mondrian: A teachable graphical editor. In *Watch what I Do*: *Programming by Demonstration*, A. Cypher (ed.). MIT Press.

LTSC. (2000) Learning technology standards committee. *http://ltsc.ieee.org/*

LTSC/IEEE LOM specification, IEEE 1484.12/D6.1 (May 2001). Available at: *http://ltsc.ieee.org/doc/wg12/LOM-WD6-1-1.pdf*

Maes, P. *et al*. (1996) The ALIVE system: Wirelless, full body interaction with autonomous agents. *ACM–Springer Multimedia Systems*, Special Issue on Multimedia and Multisensory Virtual Worlds. Springer.

Manouselis, N. and Sampson, D. (2002) Dynamic knowledge route selection for personalised learning environments using multiple criteria. In *Proceedings of the IASTED International Conference on Applied Informatics*, February, pp. 351–605.

Maulsby, D. (1993) The turvy experience: Simulating an instructible interface. In *Watch what I Do*: *Programming by Demonstration*, A. Cypher (ed.). MIT Press.

Najjar, L. J. (1996) Multimedia information and learning. *Journal of Educational Multimedia and Hypermedia*, **5**, 129–150.

Patel, A. and Kinshuk, K. (1998) Discipline attributes and teaching styles: Environmental context of an its design for determining multimedia and virtual reality representations. *Knowledge Transfer*, **1**(1), 107–113.

Rickel, J. and Johnson, W. (1998) STEVE: A pedagogical agent for virtual reality. In *Proceedings of the 2nd ACM International Conference on Autonomous Agents*, Minneapolis, USA.

Rodriguez, O. *et al.* (2002) Open learning objects: The case of inner metadata. In *Proceedings of the 11th World Wide Web Conference – WWW'2002*, Honolulu, May.

Teixeira, C., Labidi, S., and Nascimento, E. (2002) Modeling the cooperative learner based on it's actions and interactions within a teaching-learning session. In *Proceedings of the 32nd ASEE/IEEE Frontiers in Education Conference*, Boston, USA, November, pp. 19–23.